Computational Neuroscience

Terrence J. Sejnowski and Tomaso Poggio, editors

Neural Nets in Electric Fish, Walter Heiligenberg, 1991.

The Computational Brain, Patricia S. Churchland and Terrence J. Sejnowski, 1992

Dynamic Biological Networks: The Stomatogastric Nervous System, edited by Ronald M. Harris-Warrick, Eve Marder, Allen I. Selverston, and Maurice Moulins, 1992

The Neurobiology of Neural Networks, edited by Daniel Gardner, 1993

Large-Scale Neuronal Theories of the Brain, edited by Christof Koch and Joel L. Davis, 1994

The Theoretical Foundations of Dendritic Function: Selected Papers of Wilfrid Rall with Commentaries, edited by Idan Segev, John Rinzel, and Gordon M. Shepherd, 1995

Models of Information Processing in the Basal Ganglia, edited by James C. Houk, Joel L. Davis, and David G. Beiser, 1995

Spikes: Exploring the Neural Code, Fred Rieke, David Warland, Rob de Ruyter van Steveninck, and William Bialek, 1997

Neurons, Networks, and Motor Behavior, edited by Paul S.G. Stein, Sten Grillner, Allen I. Selverston, and Douglas G. Stuart, 1997

Methods in Neuronal Modeling: From Ions to Networks, second edition, edited by Christof Koch and Idan Segev, 1998

Fundamentals of Neural Network Modeling: Neuropsychology and Cognitive Neuroscience, edited by Randolph W. Parks, Daniel S. Levine, and Debra L. Long, 1998

Neural Codes and Distributed Representations: Foundations of Neural Computation, edited by Laurence Abbott and Terrence J. Sejnowski, 1998

Unsupervised Learning: Foundations of Neural Computation, edited by Geoffrey Hinton and Terrence J. Sejnowski, 1998

Fast Oscillations in Cortical Circuits, Roger D. Traub, John G.R. Jeffreys, and Miles A. Whittington, 1999

Computational Vision: Information Processing in Perception and Visual Behavior, Hanspeter A. Mallot, 2000

Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems, Peter Dayan and L.F. Abbott, 2001

First MIT Press paperback edition, 2005

Typeset in Palatino by the authors using LaTeX 2_ε.
Printed and bound in the United States of America.

Library of Congress Cataloging-in-Publication Data

Dayan, Peter.
Theoretical neuroscience : computational and mathematical modeling of neural systems / Peter Dayan and L.F. Abbott.
p. cm. – (Computational neuroscience)
Includes bibliographical references.
ISBN 0-262-04199-5 (hc. : alk. paper) — 0-262-54185-8 (pb.)
1. Neural networks (Neurobiology) – Computer simulation. 2. Human information processing – Computer simulation. 3. Computational neuroscience. I. Abbott, L.F. II. Title. III. Series

QP363.3 .D39 2001
573.8'01'13--dc21

2001044005

10 9 8 7 6 5 4

formation by populations of neurons with selective responses. Modeling of neurons and neural circuits on the basis of cellular and synaptic biophysics is presented in part II, Neurons and Neural Circuits (chapters 5–7). The role of plasticity in development and learning is discussed in part III, Adaptation and Learning (chapters 8-10). With the exception of chapters 5 and 6, which jointly cover neuronal modeling, the chapters are largely independent and can be selected and ordered in a variety of ways for a one- or two-semester course at either the undergraduate or the graduate level.

background

Although we provide some background material, readers without previous exposure to neuroscience should refer to a neuroscience textbook such as Kandel, Schwartz, & Jessell (2000); Nicholls, Martin, & Wallace (1992); Bear, Connors, & Paradiso (1996); Shepherd (1997); Zigmond et al. (1998); or Purves et al. (2000).

Theoretical neuroscience is based on the belief that methods of mathematics, physics, and computer science can provide important insights into nervous system function. Unfortunately, mathematics can sometimes seem more of an obstacle than an aid to understanding. We have not hesitated to employ the level of analysis needed to be precise and rigorous. At times, this may stretch the tolerance of some of our readers. We encourage such readers to consult the Mathematical Appendix, which provides a brief review of most of the mathematical methods used in the text, but also to persevere and attempt to understand the implications and consequences of a difficult derivation even if its steps are unclear.

exercises

Theoretical neuroscience, like any skill, can be mastered only with practice. Exercises are provided for this purpose on the web site for this book, `http://mitpress.mit.edu/dayan-abbott`. We urge the reader to do them. In addition, it will be highly instructive for the reader to construct the models discussed in the text and explore their properties beyond what we have been able to do in the available space.

Referencing

In order to maintain the flow of the text, we have kept citations within the chapters to a minimum. Each chapter ends with an annotated bibliography containing suggestions for further reading (which are denoted by a bold font), information about works cited within the chapter, and references to related studies. We concentrate on introducing the basic tools of computational neuroscience and discussing applications that we think best help the reader to understand and appreciate them. This means that a number of systems where computational approaches have been applied with significant success are not discussed. References given in the annotated bibliographies lead the reader toward such applications. Many people have contributed significantly to the areas we cover. The books and review articles in the annotated bibliographies provide more comprehensive references to work that we have failed to cite.

axons and dendrites

the dendritic tree allows a neuron to receive inputs from many other neurons through synaptic connections. The cortical pyramidal neuron of figure 1.1A and the cortical interneuron of figure 1.1C each receive thousands of synaptic inputs, and for the cerebellar Purkinje cell of figure 1.1B the number is over 100,000. Figure 1.1 does not show the full extent of the axons of these neurons. Axons from single neurons can traverse large fractions of the brain or, in some cases, of the entire body. In the mouse brain, it has been estimated that cortical neurons typically send out a total of about 40 mm of axon and have approximately 4 mm of total dendritic cable in their branched dendritic trees. The axon makes an average of 180 synaptic connections with other neurons per mm of length and the dendritic tree receives, on average, 2 synaptic inputs per μm. The cell body or soma of a typical cortical neuron ranges in diameter from about 10 to 50 μm.

ion channels

Along with these morphological features, neurons have physiological specializations. Most prominent among these are a wide variety of membrane-spanning ion channels that allow ions, predominantly sodium (Na^+), potassium (K^+), calcium (Ca^{2+}), and chloride (Cl^-), to move into and out of the cell. Ion channels control the flow of ions across the cell membrane by opening and closing in response to voltage changes and to both internal and external signals.

membrane potential

The electrical signal of relevance to the nervous system is the difference in electrical potential between the interior of a neuron and the surrounding extracellular medium. Under resting conditions, the potential inside the cell membrane of a neuron is about -70 mV relative to that of the surrounding bath (which is conventionally defined to be 0 mV), and the cell is said to be polarized. Ion pumps located in the cell membrane maintain concentration gradients that support this membrane potential difference. For example, Na^+ is much more concentrated outside a neuron than inside it, and the concentration of K^+ is significantly higher inside the neuron than in the extracellular medium. Ions thus flow into and out of a cell due to both voltage and concentration gradients. Current in the form of positively charged ions flowing out of the cell (or negatively charged ions flowing into the cell) through open channels makes the membrane potential more negative, a process called hyperpolarization. Current flowing into the cell changes the membrane potential to less negative or even positive values. This is called depolarization.

hyperpolarization and depolarization

action potential

refractory period

If a neuron is depolarized sufficiently to raise the membrane potential above a threshold level, a positive feedback process is initiated, and the neuron generates an action potential. An action potential is a roughly 100 mV fluctuation in the electrical potential across the cell membrane that lasts for about 1 ms (figure 1.2A). Action potential generation also depends on the recent history of cell firing. For a few milliseconds just after an action potential has been fired, it may be virtually impossible to initiate another spike. This is called the absolute refractory period. For a longer interval known as the relative refractory period, lasting up to tens of milliseconds after a spike, it is more difficult to evoke an action potential.

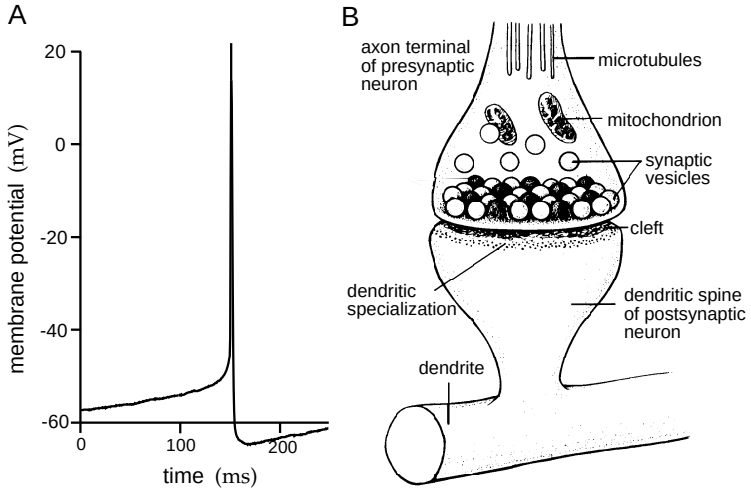

Figure 1.2 (A) An action potential recorded intracellularly from a cultured rat neocortical pyramidal cell. (B) Diagram of a synapse. The axon terminal or bouton is at the end of the axonal branch seen entering from the top of the figure. It is filled with synaptic vesicles containing the neurotransmitter that is released when an action potential arrives from the presynaptic neuron. Transmitter crosses the synaptic cleft and binds to receptors on the dendritic spine, a process roughly 1 μm long that extends from the dendrite of the postsynaptic neuron. Excitatory synapses onto cortical pyramidal cells form on dendritic spines as shown here. Other synapses form directly on the dendrites, axon, or soma of the postsynaptic neuron. (A recorded by L. Rutherford in the laboratory of G. Turrigiano; B adapted from Kandel et al., 1991.)

causing ion-conducting channels to open. Depending on the nature of the ion flow, the synapses can have either an excitatory, depolarizing, or an inhibitory, typically hyperpolarizing, effect on the postsynaptic neuron.

Recording Neuronal Responses

Figure 1.3 illustrates intracellular and extracellular methods for recording neuronal responses electrically (they can also be recorded optically). Membrane potentials are measured intracellularly by connecting a hollow glass electrode filled with a conducting electrolyte to a neuron, and comparing the potential it records with that of a reference electrode placed in the extracellular medium. Intracellular recordings are made either with sharp electrodes inserted through the membrane into the cell, or patch electrodes that have broader tips and are sealed tightly to the surface of the membrane. After the patch electrode seals, the membrane beneath its tip is either broken or perforated, providing electrical contact with the interior of the cell. The top trace in figure 1.3 is a schematic of an intracellular recording from the soma of a neuron firing a sequence of action potentials. The recording shows rapid spikes riding on top of a more slowly varying subthreshold potential. The bottom trace is a schematic of an intracellular recording made some distance out on the axon of the neuron. These traces

sharp and patch electrodes

Index

bold fonts mark sections or subsections in the text
italic fonts mark margin labels

From Stimulus to Response

Characterizing the relationship between stimulus and response is difficult because neuronal responses are complex and variable. Neurons typically respond by producing complex spike sequences that reflect both the intrinsic dynamics of the neuron and the temporal characteristics of the stimulus. Isolating features of the response that encode changes in the stimulus can be difficult, especially if the time scale for these changes is of the same order as the average interval between spikes. Neural responses can vary from trial to trial even when the same stimulus is presented repeatedly. There are many potential sources of this variability, including variable levels of arousal and attention, randomness associated with various biophysical processes that affect neuronal firing, and the effects of other cognitive processes taking place during a trial. The complexity and trial-to-trial variability of action potential sequences make it unlikely that we can describe and predict the timing of each spike deterministically. Instead, we seek a model that can account for the probabilities that different spike sequences are evoked by a specific stimulus.

Typically, many neurons respond to a given stimulus, and stimulus features are therefore encoded by the activities of large neural populations. In studying population coding, we must examine not only the firing patterns of individual neurons but also the relationships of these firing patterns to each other across the population of responding cells.

In this chapter, we introduce the firing rate and spike-train correlation functions, which are basic measures of spiking probability and statistics. We also discuss spike-triggered averaging, a method for relating action potentials to the stimulus that evoked them. Finally, we present basic stochastic descriptions of spike generation, the homogeneous and inhomogeneous Poisson models, and discuss a simple model of neural responses to which they lead. In chapter 2, we continue our discussion of neural encoding by showing how reverse-correlation methods are used to construct estimates of firing rates in response to time-varying stimuli. These methods have been applied extensively to neural responses in the retina, lateral geniculate nucleus (LGN) of the thalamus, and primary visual cortex, and we review the resulting models.

1.2 Spike Trains and Firing Rates

Action potentials convey information through their timing. Although action potentials can vary somewhat in duration, amplitude, and shape, they are typically treated as identical stereotyped events in neural encoding studies. If we ignore the brief duration of an action potential (about 1 ms), an action potential sequence can be characterized simply by a list of the times when spikes occurred. For n spikes, we denote these times by t_i with $i = 1, 2, \ldots, n$. The trial during which the spikes are recorded is taken to

Troyer, TW, & Miller, KD (1997) Physiological gain leads to high ISI variability in a simple model of a cortical regular spiking cell. *Neural Computation* **9**:971– 983.

Tsai, KY, Carnevale, NT, Claiborne, BJ, & Brown, TH (1994) Efficient mapping from neuroanatomical to electrotonic space. *Network: Computation in Neural Systems* **5**:21–46.

Tsodyks, MV, & Markram, H (1997) The neural code between neocortical pyramidal neurons depends on neurotransmitter release probability. *Proceedings of the National Academy of Sciences of the United States of America* **94**:719–723.

Tuckwell, HC (1988) *Introduction to Theoretical Neurobiology*. Cambridge, UK: Cambridge University Press.

Turing, AM (1952) The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society of London* **B237**:37–72.

Turrigiano, G, LeMasson, G, & Marder, E (1995) Selective regulation of current densities underlies spontaneous changes in the activity of cultured neurons. *Journal of Neuroscience* **15**:3640–3652.

Uttley, AM (1979) *Information Transmission in the Nervous System*. London: Academic Press.

Van Essen, DC, Newsome, WT, & Maunsell, JHR (1984) The visual field representation in striate cortex of the macaque monkey: Asymmetries, anisotropies, and individual variability. *Vision Research* **24**:429–448.

Van Santen, JP, & Sperling, G (1984) Temporal covariance model of human motion perception. *Journal of the Optical Society of America* **A1**:451–473.

Varela, J, Sen, K, Gibson, J, Fost, J, Abbott, LF, Nelson, SB (1997) A quantitative description of short-term plasticity at excitatory synapses in layer 2/3 of rat primary visual cortex. *Journal of Neuroscience* **17**:7926–7940.

Venkatesh, SS (1986) Epsilon capacity of a neural network. In J Denker, ed., *Proceedings of Neural Networks for Computing.* AIP Conference Proceedings Volume 151, New York: American Institute of Physics, 440–445.

Vogels, R (1990) Population coding of stimulus orientation by cortical cells. *Journal of Neuroscience* **10**:3543–3558.

van Vreeswijk, C, Abbott, LF, & Ermentrout, GB (1994) When inhibition not excitation synchronizes neuronal firing. *Journal of Computational Neuroscience* **1**:313–321.

Wallis, G, & Baddeley, R (1997) Optimal, unsupervised learning in invariant object recognition. *Neural Computation* **9**:883–894.

Wandell, BA (1995) *Foundations of Vision*. Sunderland, MA: Sinauer Associates.

Wang, X-J (1994) Multiple dynamical modes of thalamic relay neurons: Rhythmic bursting and intermittent phase-locking. *Neuroscience* **59**:21–31.

Wang, X-J (1998) Calcium coding and adaptive temporal computation in cortical pyramidal neurons. *Journal of Neurophysiology* **79**:1549–1566.

Wang, X-J, & Rinzel, J (1992) Alternating and synchronous rhythms in reciprocally inhibitory model neurons. *Neural Computation* **4**:84–97.

Watkins, CJCH (1989) *Learning from Delayed Rewards*. Ph.D. dissertation, University of Cambridge.

Watson, AB, & Ahumada, AJ (1985) Model of human visual-motion sensing. *Journal of the Optical Society of America* **A2**:322–342.

as the average number of spikes (averaged over trials) appearing during a short interval between times t and $t + \Delta t$, divided by the duration of the interval.

trial average $\langle\ \rangle$

The number of spikes occurring between times t and $t + \Delta t$ on a single trial is the integral of the neural response function over that time interval. The average number of spikes during this interval is the integral of the trial-averaged neural response function. We use angle brackets, $\langle\ \rangle$, to denote averages over trials that use the same stimulus, so that $\langle z \rangle$ for any quantity z is the sum of the values of z obtained from many trials involving the same stimulus, divided by the number of trials. The trial-averaged neural response function is denoted by $\langle \rho(t) \rangle$, and the time-dependent firing rate is given by

firing rate $\mathsf{r}(t)$

$$\mathsf{r}(t) = \frac{1}{\Delta t} \int_t^{t+\Delta t} d\tau \, \langle \rho(\tau) \rangle \, . \tag{1.5}$$

We use the notation $\mathsf{r}(t)$ for this important quantity (as opposed to r for the spike-count rate), and when we use the term "firing rate" without any modifiers, we mean $\mathsf{r}(t)$. Formally, the limit $\Delta t \to 0$ should be taken on the right side of this expression, but, in extracting a time-dependent firing rate from data, the value of Δt must be large enough so there are sufficient numbers of spikes within the interval defining $\mathsf{r}(t)$ to obtain a reliable estimate of the average.

For sufficiently small Δt, $\mathsf{r}(t)\,\Delta t$ is the average number of spikes occurring between times t and $t + \Delta t$ over multiple trials. The average number of spikes over a longer time interval is given by the integral of $\mathsf{r}(t)$ over that interval. If Δt is small, there will never be more than one spike within the interval between t and $t + \Delta t$ on any given trial. This means that $\mathsf{r}(t)\,\Delta t$ is also the fraction of trials on which a spike occurred between those times. Equivalently, $\mathsf{r}(t)\,\Delta t$ is the probability that a spike occurs during this time interval. This probabilistic interpretation provides a formal definition of the time-dependent firing rate; $\mathsf{r}(t)\,\Delta t$ is the probability of a spike occurring during a short interval of duration Δt around the time t.

spiking probability

In any integral expression such as equation 1.2, the neural response function generates a contribution whenever a spike occurs. If we use the trial-average response function instead, as in equation 1.5, this generates contributions proportional to the fraction of trials on which a spike occurred. Because of the relationship between this fraction and the firing rate, we can replace the trial-averaged neural response function with the firing rate $\mathsf{r}(t)$ within any well-behaved integral, for example,

$$\int d\tau \, h(\tau) \, \langle \rho(t-\tau) \rangle = \int d\tau \, h(\tau) \mathsf{r}(t-\tau) \tag{1.6}$$

for any function h. This establishes an important relationship between the average neural response function and the firing rate; the two are equivalent when used inside integrals. It also provides another interpretation of $\mathsf{r}(t)$ as the trial-averaged density of spikes along the time axis.

Sejnowski, TJ (1977) Storing covariance with nonlinearly interacting neurons. *Journal of Mathematical Biology* **4**:303–321.

Sejnowski, TJ (1999) The book of Hebb. *Neuron* **24**:773–776.

Seung, HS (1996) How the brain keeps the eyes still. *Proceedings of the National Academy of Sciences of the United States of America* **93**:13339–13344.

Seung, HS, Lee, DD, Reis, BY, & Tank DW (2000) Stability of the memory for eye position in a recurrent network of conductance-based model neurons. *Neuron* **26**:259–271.

Seung, HS, & Sompolinsky, H (1993) Simple models for reading neuronal population codes. *Proceedings of the National Academy of Sciences of the United States of America* **90**:10749–10753.

Shadlen, MN, Britten, KH, Newsome, WT, & Movshon, JA (1996) A computational analysis of the relationship between neuronal and behavioral responses to visual motion. *Journal of Neuroscience* **16**:1486–1510.

Shadlen, MN, & Newsome WT (1998) The variable discharge of cortical neurons: Implications for connectivity, computation, and information coding. *Journal of Neuroscience* **18**:3870–3896.

Shanks, DR (1995) *The Psychology of Associative Learning*. Cambridge: Cambridge University Press.

Shannon, CE, & Weaver, W (1949) *The Mathematical Theory of Communications*. Urbana, IL: University of Illinois Press.

Shepherd, GM (1997) *Neurobiology*. Oxford: Oxford University Press.

Siebert, WMcC (1986) *Circuits, Signals, and Systems*. Cambridge, MA: MIT Press; New York: McGraw-Hill.

Simoncelli, EP (1997) Statistical models for images: Compression, restoration and synthesis. *Proceedings of the 31st Asilomar Conference on Signals, Systems and Computers*. Pacific Grove, CA: IEEE Computer Society, 673–678.

Simoncelli, EP, & Adelson, EH (1990) Subband image coding. In JW Woods, ed., *Subband Transforms.*, 143–192 Norwell, MA: Kluwer Academic Publishers.

Simoncelli, EP, & Freeman, WT (1995) The steerable pyramid: A flexible architecture for derivative computation. *Proceedings of the International Conference on Image Processing*, 444–447. Los Alamitos, CA: IEEE Computer Society Press.

Simoncelli, EP, Freeman, WT, Adelson, EH, & Heeger, DJ (1992) Shiftable multiscale transforms. *IEEE Transactions on Information Theory* **38**:587–607.

Simoncelli, EP, & Schwartz, O (1999) Modeling non-specific suppression in V1 neurons with a statistically-derived normalization model. In MS Kearns, SA Solla, & DA Cohn, eds. *Advances in Neural Information Processing Systems, 11*, 153–159. Cambridge, MA: MIT Press.

Snippe, HP (1996) Parameter extraction from population codes: A critical assessment. *Neural Computation* **8**:511–529.

Snippe, HP & Koenderink, JJ (1992a) Discrimination thresholds for channel-coded systems. *Biological Cybernetics* **66**:543-551.

Snippe, HP, & Koenderink, JJ (1992b) Information in channel-coded systems: Correlated receivers. *Biological Cybernetics* **67**:183–190.

Softky, WR, & Koch, C (1992) Cortical cells should spike regularly but do not. *Neural Computation* **4**:643–646.

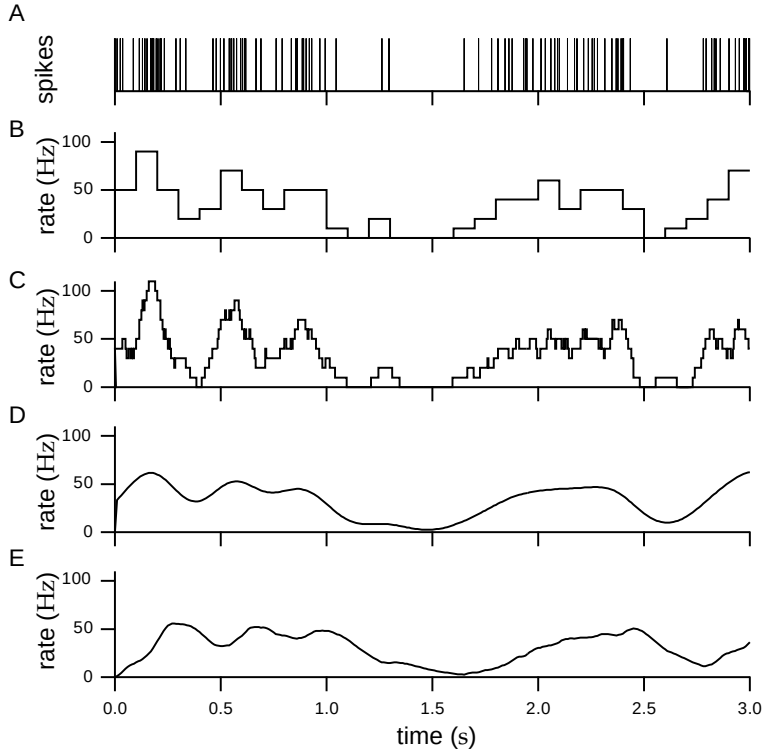

Figure 1.4 Firing rates approximated by different procedures. (A) A spike train from a neuron in the inferotemporal cortex of a monkey recorded while that animal watched a video on a monitor under free viewing conditions. (B) Discrete-time firing rate obtained by binning time and counting spikes with $\Delta t = 100$ ms. (C) Approximate firing rate determined by sliding a rectangular window function along the spike train with $\Delta t = 100$ ms. (D) Approximate firing rate computed using a Gaussian window function with σ_t = 100 ms. (E) Approximate firing rate using the window function of equation 1.12 with $1/\alpha = 100$ ms. (Data from Baddeley et al., 1997.)

The binning and counting procedure illustrated in figure 1.4B generates an estimate of the firing rate that is a piecewise constant function of time, resembling a histogram. Because spike counts can take only integer values, the rates computed by this method will always be integer multiples of $1/\Delta t$, and thus they take discrete values. Decreasing the value of Δt increases temporal resolution by providing an estimate of the firing rate at more finely spaced intervals of time, but at the expense of decreasing the resolution for distinguishing different rates. One way to avoid quantized firing rates is to vary the bin size so that a fixed number of spikes appears in each bin. The firing rate is then approximated as that fixed number of spikes divided by the variable bin width.

Counting spikes in preassigned bins produces a firing-rate estimate that depends not only on the size of the time bins but also on their placement. To avoid the arbitrariness in the placement of bins, we can instead take a single bin or window of duration Δt and slide it along the spike train,

Pouget, A, Zhang, KC, Deneve, S, & Latham, PE (1998) Statistically efficient estimation using population coding. *Neural Computation* **10**:373–401.

Press, WH, Teukolsky, SA, Vetterling, WT, & Flannery, BP (1992) *Numerical Recipes in C*. Cambridge: Cambridge University Press.

Price, DJ, & Willshaw, DJ (2000) *Mechanisms of Cortical Development*. Oxford: Oxford University Press.

Purves, D, Augustine, GJ, Fitzpatrick, D, Katz, LC, LaMantia, A-S, McNamara, JO, & Williams, SM, eds. (2000) *Neuroscience*. Sunderland MA: Sinauer Associates.

Rall, W (1959) Branching dendritic trees and motoneuron membrane resistivity. *Experimental Neurology* **2**:503–532.

Rall, W (1977) Core conductor theory and cable properties of neurons. In Kandel, ER, ed., *Handbook of Physiology*, vol. 1, 39–97. Bethesda, MD: American Physiology Society.

Rao, RPN, & Ballard, DH (1997) Dynamic model of visual recognition predicts neural response properties in the visual cortex. *Neural Computation* **9**:721–763.

Raymond, JL, Lisberger, SG, & Mauk, MD (1996) The cerebellum: A neuronal learning machine? *Science* **272**:1126–1131.

Real, LA (1991) Animal choice behavior and the evolution of cognitive architecture. *Science* **253**:980–986.

Reichardt, W (1961) Autocorrelation: A principle for the evaluation of sensory information by the central nervous system. In WA Rosenblith, ed., *Sensory Communication*. New York: Wiley.

Rescorla, RA, & Wagner, AR (1972) A theory of Pavlovian conditioning: The effectiveness of reinforcement and non-reinforcement. In AH Black & WF Prokasy, eds., *Classical Conditioning II: Current Research and Theory*, 64–69. New York: Appleton-Century-Crofts.

Rieke F, Bodnar, DA, & Bialek, W (1995) Naturalistic stimuli increase the rate and efficiency of information transmission by primary auditory afferents. *Proceedings of the Royal Society of London*. **B262**:259–265.

Rieke, FM, Warland, D, de Ruyter van Steveninck, R, & Bialek, W (1997) *Spikes: Exploring the Neural Code*. Cambridge, MA: MIT Press.

Rinzel, J, & Ermentrout, B (1998) Analysis of neural excitability and oscillations. In C Koch & I Segev, eds., *Methods in Neuronal Modeling: From Synapses to Networks*, 251–292. Cambridge, MA: MIT Press.

Rissanen, J (1989) *Stochastic Complexity in Statistical Inquiry*. Singapore: World Scientific Press.

Robinson, DA (1989) Integrating with neurons. *Annual Review of Neuroscience* **12**:33–45.

Rodieck, R (1965) Quantitative analysis of cat retinal ganglion cell responses to visual stimuli. *Vision Research* **5**:583–601.

Rolls, ET, & Treves, A (1998) *Neural Networks and Brain Function*. New York: Oxford University Press.

Rosenblatt, F (1958) The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review* **65**:386–408.

Roth, Z, & Baram, Y (1996) Multidimensional density shaping by sigmoids. *IEEE Transactions on Neural Networks* **7**:1291–1298.

Rovamo, J, & Virsu, V (1984) Isotropy of cortical magnification and topography of striate cortex. *Vision Research* **24**:283–286.

is negative. Such a window function or kernel is called causal. One commonly used form is the α function

$$w(\tau) = [\alpha^2 \tau \exp(-\alpha\tau)]_+ \tag{1.12}$$

half-wave rectification $[\,]_+$

where $1/\alpha$ determines the temporal resolution of the resulting firing-rate estimate. The notation $[z]_+$ for any quantity z stands for the half-wave rectification operation,

$$[z]_+ = \begin{cases} z & \text{if } z \geq 0 \\ 0 & \text{otherwise}. \end{cases} \tag{1.13}$$

Figure 1.4E shows the firing rate approximated by such a causal scheme. Note that this rate tends to peak later than the rate computed in figure 1.4D using a temporally symmetric window function.

Tuning Curves

stimulus s

Neuronal responses typically depend on many different properties of a stimulus. In this chapter, we characterize responses of neurons as functions of just one of the stimulus attributes to which they may be sensitive. The value of this single attribute is denoted by s. In chapter 2, we consider more complete stimulus characterizations.

response tuning curve $f(s)$

A simple way of characterizing the response of a neuron is to count the number of action potentials fired during the presentation of a stimulus. This approach is most appropriate if the parameter s characterizing the stimulus is held constant over the trial. If we average the number of action potentials fired over (in theory, an infinite number of) trials and divide by the trial duration, we obtain the average firing rate, $\langle r \rangle$, defined in equation 1.7. The average firing rate written as a function of s, $\langle r \rangle = f(s)$, is called the neural response tuning curve. The functional form of a tuning curve depends on the parameter s used to describe the stimulus. The precise choice of parameters used as arguments of tuning curve functions is partially a matter of convention. Because tuning curves correspond to firing rates, they are measured in units of spikes per second or Hz.

primary visual cortex V1

Figure 1.5A shows extracellular recordings of a neuron in the primary visual cortex (V1) of a monkey. While these recordings were being made, a bar of light was moved at different angles across the region of the visual field where the cell responded to light. This region is called the receptive field of the neuron. Note that the number of action potentials fired depends on the angle of orientation of the bar. The same effect is shown in figure 1.5B in the form of a response tuning curve, which indicates how the average firing rate depends on the orientation of the light bar stimulus.

Gaussian tuning curve

The data have been fitted by a response tuning curve of the form

$$f(s) = r_{\max} \exp\left(-\frac{1}{2}\left(\frac{s - s_{\max}}{\sigma_f}\right)^2\right), \tag{1.14}$$

Movshon JA, Thompson ID, & Tolhurst DJ (1978a) Spatial summation in the receptive fields of simple cells in the cat's striate cortex. *Journal of Neurophysiology* **283**:53–77.

Movshon JA, Thompson ID, & Tolhurst DJ (1978b) Spatial and temporal contrast sensitivity of neurones in areas 17 and 18 of the cat's visual cortex. *Journal of Neurophysiology* **283**:101–120.

Mumford, D (1994) Neuronal architectures for pattern-theoretic problems. In C Koch, & J Davis, eds., *Large-Scale Theories of the Cortex*, 125–152. Cambridge, MA: MIT Press.

Murray, JD (1993) *Mathematical Biology*. New York: Springer-Verlag.

Narendra, KS, & Thatachar, MAL (1989) *Learning Automata: An Introduction*. Englewood Cliffs, NJ: Prentice-Hall.

Neal, RM (1993) *Probabilistic Inference Using Markov Chain Monte Carlo Methods*. Department of Computer Science, University of Toronto, technical teport CRG-TR-93-1.

Neal, RM, & Hinton, GE (1998) A view of the EM algorithm that justifies incremental, sparse, and other variants. In MI Jordan, ed., *Learning in Graphical Models*, 355–368. Dordrecht: Kluwer Academic Publishers.

Neisser, U (1967) *Cognitive Psychology*. New York: Appleton-Century-Crofts.

Newsome, WT, Britten, KH, & Movshon, JA (1989) Neural correlates of a perceptual decision. *Nature* **341**:52–54.

Nicholls, JG, Martin, R, & Wallace, BG (1992) *From Neuron to Brain: A Cellular and Molecular Approach to the Function of the Nervous System*. Sunderland, MA: Sinauer Associates.

Nowlan, SJ (1991) *Soft Competitive Adaptation: Neural Network Learning Algorithms Based on Fitting Statistical Mixtures*. Ph.D. dissertation, Carnegie-Mellon University.

Obermayer, K, & Blasdel, GG (1993) Geometry of orientation and ocular dominance columns in monkey striate cortex. *Journal of Neuroscience* **13**:4114–4129.

Obermayer, K, Blasdel, GG, & Schulten, K (1992) Statistical-mechanical analysis of self-organization and pattern formation during the development of visual maps. *Physical Review A* **45**:7568–7589.

Oja, E (1982) A simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology* **16**:267–273.

O'Keefe, J, & Recce, ML (1993) Phase relationship between hippocampal place units and the EEG theta rhythm. *Hippocampus* **3**:317–330.

O'Keefe, LP, Bair, W, & Movshon, JA (1997) Response variability of MT neurons in macaque monkey. *Society for Neuroscience Abstracts* **23**:1125.

Olshausen, B (1996) *Learning Linear, Sparse, Factorial Codes*. MIT AI Lab, MIT, AI-memo 1580.

Olshausen, BA, & Field, DJ (1996) Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature* **381**:607–609.

Olshausen, BA, & Field, DJ (1997) Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research* **37**:3311–3325.

Oppenheim, AV, & Willsky, AS, with Nawab, H (1997) *Signals and Systems*. 2nd ed. Upper Saddle River, NJ: Prentice-Hall.

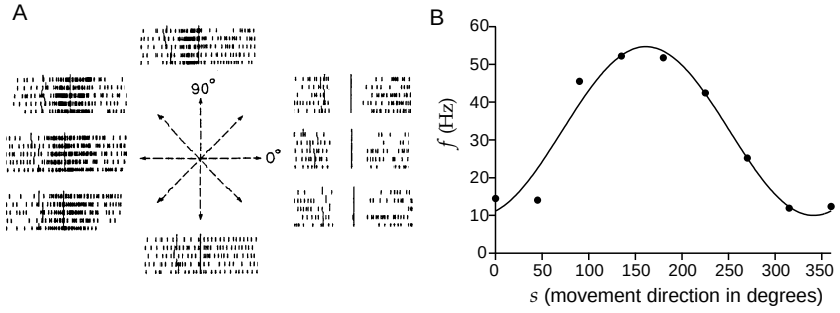

Figure 1.6 (A) Recordings from the primary motor cortex of a monkey performing an arm-reaching task. The hand of the monkey started from a central resting location, and reaching movements were made in the directions indicated by the arrows. The rasters for each direction show action potentials fired on five trials. (B) Average firing rate plotted as a function of the direction in which the monkey moved its arm. The curve is a fit using the function 1.15 with parameters $r_{\max} = 54.69$ Hz, $r_0 = 32.34$ Hz, and $s_{\max} = 161.25^\circ$. (A adapted from Georgopoulos et al., 1982, which is also the source of the data points in B.)

ated with the maximum response $r_{\max}$, and r_0 is an offset or background firing rate that shifts the tuning curve up from the zero axis. The minimum firing rate predicted by equation 1.15 is $2r_0 - r_{\max}$. For the neuron of figure 1.6B, this is a positive quantity, but for some M1 neurons $2r_0 - r_{\max} < 0$, and the function 1.15 is negative over some range of angles. Because firing rates cannot be negative, the cosine tuning curve must be half-wave rectified in these cases (see equation 1.13),

$$f(s) = [r_0 + (r_{\max} - r_0)\cos(s - s_{\max})]_+ . \tag{1.16}$$

Figure 1.7B shows how the average firing rate of a V1 neuron depends on retinal disparity and illustrates another important type of tuning curve. Retinal disparity is a difference in the retinal location of an image between the two eyes (figure 1.7A). Some neurons in area V1 are sensitive to disparity, representing an early stage in the representation of viewing distance. In figure 1.7B, the data points have been fitted with a tuning curve called a logistic or sigmoidal function,

sigmoidal tuning curve

$$f(s) = \frac{r_{\max}}{1 + \exp\left((s_{1/2} - s)/\Delta_s\right)} . \tag{1.17}$$

In this case, s is the retinal disparity, the parameter $s_{1/2}$ is the disparity that produces a firing rate half as big as the maximum value $r_{\max}$, and Δ_s controls how quickly the firing rate increases as a function of s. If Δ_s is negative, the firing rate is a monotonically decreasing function of s rather than a monotonically increasing function as in figure 1.7B.

Spike-Count Variability

Tuning curves allow us to predict the average firing rate, but they do not describe how the spike-count firing rate r varies about its mean value

Li, Z (1998) A neural model of contour integration in the primary visual cortex. *Neural Computation* **10**:903–940.
Li, Z (1999) Visual segmentation by contextual influences via intra-cortical interactions in the primary visual cortex. *Network: Computation in Neural Systems* **10**:187–212.
Li, Z, & Atick, JJ (1994a) Efficient stereo coding in the multiscale representation. *Network: Computation in Neural Systems* **5**:157–174.
Li, Z, & Atick, JJ (1994b) Toward a theory of the striate cortex. *Neural Computation* **6**:127–146.
Li, Z, & Dayan, P (1999) Computational differences between asymmetrical and symmetrical networks. *Network: Computation in Neural Systems* **10**:59–78.
Li, Z, & Hopfield, JJ (1989) Modeling the olfactory bulb and its neural oscillatory processings. *Biological Cybernetics* **61**:379–392.
Lindgren, BW (1993) *Statistical Theory*. 4th ed. New York: Chapman & Hall.
Linsker, R (1986) From basic network principles to neural architecture. *Proceedings of the National Academy of Sciences of the United States of America* **83**:7508–7512, 8390–8394, 8779–8783.
Linsker, R (1988) Self-organization in a perceptual network. *Computer* **21**:105–117.
Lisman, JE (1997) Bursts as a unit of neural information: making unreliable synapses reliable. *Trends in Neuroscience* **20**:38–43.
Liu, Z, Golowasch, J, Marder, E, & Abbott, LF (1998) A model neuron with activity-dependent conductances regulated by multiple calcium sensors. *Journal of Neuroscience* **18**:2309–2320.
MacKay, DJC (1996) *Maximum Likelihood and Covariant Algorithms for Independent Components Analysis.* Unpublished manuscript.
MacKay, DJC, & Miller, KD (1990) Analysis of Linsker's application of Hebbian rules to linear networks. *Network: Computation in Neural Systems* **1**:257–299.
MacKay, DM (1956) The epistemological problem for automata. In CE Shannon, & J McCarthy, eds., *Automata Studies*, 235–251. Princeton, NJ: Princeton University Press.
Mackintosh, NJ (1983) *Conditioning and Associative Learning*. Oxford: Oxford University Press.
Magleby, KL (1987) Short-term changes in synaptic efficacy. In G Edelman, W Gall, & W Cowan, eds., *Synaptic Function*, 21–56. New York: Wiley.
Mangel, M, & Clark, CW (1988) *Dynamic Modeling in Behavioral Ecology*. Princeton, NJ: Princeton University Press.
Mallat, SG (1998) *A Wavelet Tour of Signal Processing*. San Diego, CA: Academic Press.
Marder, E, & Calabrese, RL (1996) Principles of rhythmic motor pattern generation. *Physiological Review* **76**:687–717.
Markram, H, Lubke, J, Frotscher, M, & Sakmann, B (1997) Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs. *Science* **275**:213–215.
Markram, H, & Tsodyks, M (1996) Redistribution of synaptic efficacy between neocortical pyramidal neurons. *Nature* **382**:807–810.
Markram, H, Wang, Y, & Tsodyks, MV (1998) Differential signalling via the same axon of neocortical pyramidal neurons. *Proceedings of the National Academy of Sciences of the United States of America* **95**:5323–5328.

eraging the stimuli that produce a given response. To average stimuli in this way, we need to specify what fixed response we will use to "trigger" the average. The most obvious choice is the firing of an action potential. Thus, we ask, "What, on average, did the stimulus do before an action potential was fired?" The resulting quantity, called the spike-triggered average stimulus, provides a useful way of characterizing neuronal selectivity. Spike-triggered averages are computed using stimuli characterized by a parameter $s(t)$ that varies over time. Before beginning our discussion of spike triggering, we describe some features of such stimuli.

Describing the Stimulus

Neurons responding to sensory stimuli face the difficult task of encoding parameters that can vary over an enormous dynamic range. For example, photoreceptors in the retina can respond to single photons or can operate in bright light with an influx of millions of photons per second. To deal with such wide-ranging stimuli, sensory neurons often respond most strongly to rapid changes in stimulus properties and are relatively insensitive to steady-state levels. Steady-state responses are highly compressed functions of stimulus intensity, typically with logarithmic or weak power-law dependences. This compression has an interesting psychophysical correlate. Weber measured how different the intensity of two stimuli had to be for them to be reliably discriminated, the "just noticeable" difference Δs. He found that, for a given stimulus, Δs is proportional to the magnitude of the stimulus s, so that $\Delta s/s$ is constant. This relationship is called Weber's law. Fechner suggested that noticeable differences set the scale for perceived stimulus intensities. Integrating Weber's law, this means that the perceived intensity of a stimulus of absolute intensity s varies as $\log s$. This is known as Fechner's law.

Weber's law

Fechner's law

Sensory systems make numerous adaptations, using a variety of mechanisms, to adjust to the average level of stimulus intensity. When a stimulus generates such adaptation, the relationship between stimulus and response is often studied in a potentially simpler regime by describing responses to fluctuations about a mean stimulus level. In this case, $s(t)$ is defined so that its time average over the duration of a trial is 0, $\int_0^T dt\, s(t)/T = 0$. We frequently impose this condition.

$\int_0^T dt\, s(t)/T = 0$

Our analysis of neural encoding involves two different types of averages: averages over repeated trials that employ the same stimulus, which we denote by angle brackets, and averages over different stimuli. We could introduce a second notation for averages over stimuli, but this can be avoided when using time-dependent stimuli. Instead of presenting a number of different stimuli and averaging over them, we can string together all of the stimuli we wish to consider into a single time-dependent stimulus sequence and average over time. Thus, stimulus averages are replaced by time averages.

stimulus and time averages

Although a response recorded over a trial depends only on the values

Hopfield, JJ (1984) Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Sciences of the United States of America* **81**:3088–3092.

Houk, JC, Adams, JL, & Barto, AG (1995) A model of how the basal ganglia generate and use neural signals that predict reinforcement. In JC Houk, JL Davis, & DG Beiser, eds., *Models of Information Processing in the Basal Ganglia*, 249–270. Cambridge, MA: MIT Press.

Houk, JC, Davis, JL, & Beiser, DG, eds. (1995) *Models of Information Processing in the Basal Ganglia*. Cambridge, MA: MIT Press.

Hubel, DH (1988) *Eye, Brain, and Vision*. New York: WH Freeman.

Hubel, DH, & Wiesel, TN (1962) Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *Journal of Physiology* **160**:106–154.

Hubel, DH, & Wiesel, TN (1968) Receptive fields and functional architecture of the monkey striate cortex. *Journal of Physiology* **195**:215–243.

Hubel, DH, & Wiesel, TN (1977) Functional architecture of macaque monkey visual cortex. *Proceedings of the Royal Society of London* **B198**:1–59.

Hubener, M, Shoham, D, Grinvald, A, & Bonhoeffer, T (1997) Spatial relationships among three columnar systems in cat area 17. *Journal of Neuroscience* **17**:9270–9284.

Huber, PJ (1985) Projection pursuit. *Annals of Statistics* **13**:435–475.

Huguenard, JR, & McCormick, DA (1992) Simulation of the currents involved in rhythmic oscillations in thalamic relay neurons. *Journal of Neurophysiology* **68**:1373–1383.

Humphrey, DR, Schmidt, EM, & Thompson, WD (1970) Predicting measures of motor performance from multiple cortical spike trains. *Science* **170**:758–761.

Intrator, N, & Cooper, LN (1992) Objective function formulation of the BCM theory of visual cortical plasticity: Statistical connections, stability conditions. *Neural Networks* **5**:3–17.

Jack, JJB, Noble, D, & Tsien, RW (1975) *Electrical Current Flow in Excitable Cells*. Oxford: Oxford University Press.

Jahr, CE, & Stevens, CF (1990) A quantitative description of NMDA receptor channel kinetic behavior. *Journal of Neuroscience* **10**:1830–1837.

Johnston, D, & Wu, SM (1995) *Foundations of Cellular Neurophysiology*. Cambridge, MA: MIT Press.

Jolliffe, IT (1986) *Principal Component Analysis*. New York: Springer-Verlag.

Jones, J, & Palmer, L (1987a) The two-dimensional spatial structure of simple receptive fields in cat striate cortex. *Journal of Neurophysiology* **58**:1187–1211.

Jones, J, & Palmer, L (1987b) An evaluation of the two-dimensional Gabor filter model of simple receptive fields in cat striate cortex. *Journal of Neurophysiology* **58**:1233–1258.

Jordan, DW, & Smith, P (1977) *Nonlinear Ordinary Differential Equations*. Oxford: Clarendon Press.

Jordan, MI, ed. (1998) *Learning in Graphical Models*. Dordrecht: Kluwer Academic Publishers.

Jordan, MI, Ghahramani, Z, Jaakkola, TS, & Saul, LK (1998). An introduction to variational methods for graphical models. In MI Jordan, ed., *Learning in Graphical Models*, 105–162. Dordrecht: Kluwer Academic Publishers.

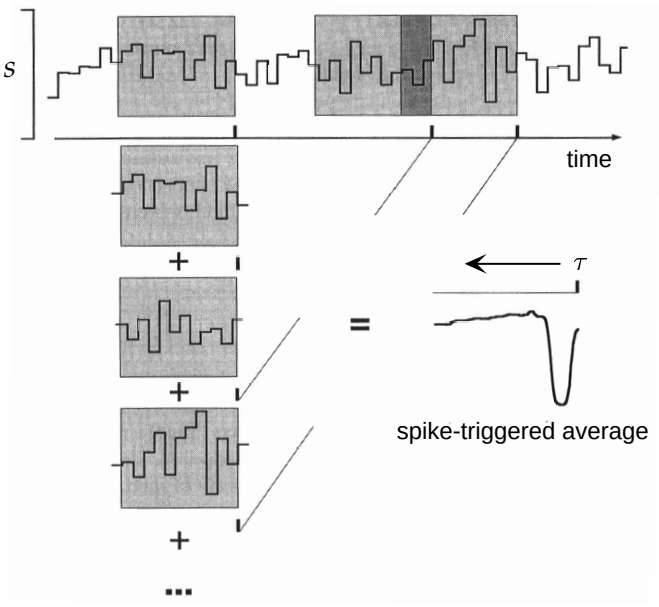

Figure 1.8 Schematic of the procedure for computing the spike-triggered average stimulus. Each gray rectangle contains the stimulus prior to one of the spikes shown along the time axis. These are averaged to produce the waveform shown at the lower right, which is the average stimulus before a spike. The stimulus in this example is a piecewise constant function of time. (Adapted from Rieke et al., 1997.)

The second equality is due to the equivalence of $\langle \rho(t) \rangle$ and $r(t)$ within integrals. Equation 1.20 allows us to relate the spike-triggered average to the correlation function of the firing rate and the stimulus.

firing-rate stimulus correlation function Q_{rs}

Correlation functions are a useful way of determining how two quantities that vary over time are related to one another. The two quantities being related are evaluated at different times, one at time t and the other at time $t + \tau$. The correlation function is then obtained by averaging their product over all t values, and it is a function of τ. The correlation function of the firing rate and the stimulus is

$$Q_{rs}(\tau) = \frac{1}{T} \int_0^T dt\, r(t) s(t + \tau)\,. \tag{1.21}$$

By comparing equations 1.20 and 1.21, we find that

$$C(\tau) = \frac{1}{\langle r \rangle} Q_{rs}(-\tau)\,, \tag{1.22}$$

reverse correlation function

where $\langle r \rangle = \langle n \rangle / T$ is the average firing rate over the set of trials. Because the argument of the correlation function in equation 1.22 is $-\tau$, the spike-triggered average stimulus is often called the reverse correlation function. It is proportional to the correlation of the firing rate with the stimulus at preceding times.

van Gisbergen, JAM, Van Opstal, AJ, & Tax, AMM (1987) Collicular ensemble coding of saccades based on vector summation. *Neuroscience* **21**:541–555.

Gluck, MA, Reifsnider, ES, & Thompson, RF (1990) Adaptive signal processing and the cerebellum: Models of classical conditioning and VOR adaptation. In MA Gluck, & DE Rumelhart, eds., *Neuroscience and Connectionist Theory. Developments in Connectionist Theory*, 131–185. Hillsdale, NJ: Erlbaum.

Gluck, MA, & Rumelhart, DE, eds. (1990) *Neuroscience and Connectionist Theory*. Hillsdale, NJ: Erlbaum.

Goldman-Rakic, PS (1994) Cellular basis of working memory. *Neuron* **14**:477–485.

Goodall, MC (1960) Performance of a stochastic net. *Nature* **185**:557–558.

Goodhill, GJ (1993) Topography and ocular dominance: A model exploring positive correlations. *Biological Cybernetics* **69**:109–118.

Goodhill, GJ, & Richards, LJ (1999) Retinotectal maps: Molecules, models and misplaced data. *Trends in Neuroscience* **22**:529–534.

Goodhill, GJ, & Willshaw, DJ (1990) Application of the elastic net algorithm to the formation of ocular dominance stripes. *Network: Computation in Neural Systems* **1**:41–61.

Graham, NVS (1989) *Visual Pattern Analyzers*. New York: Oxford University Press.

Graziano, MSA, Hu, XT, & Gross, CG (1997) Visuospatial properties of ventral premotor cortex. *Journal of Neurophysiology* **77**:2268–2292.

Green, DM, & Swets, JA (1966) *Signal Detection Theory and Psychophysics*. Los Altos, CA: Peninsula Publishing.

Grenander, U (1995) *Elements of Pattern Theory*. Baltimore: Johns Hopkins University Press.

Grossberg S (1982) Processing of expected and unexpected events during conditioning and attention: A psychophysiological theory. *Psychological Review* **89**:529–572.

Grossberg, S, ed. (1987) *The Adaptive Brain. Vols. 1 and 2*. Amsterdam: Elsevier.

Grossberg, S, ed. (1988) *Neural Networks and Natural Intelligence*. Cambridge, MA: MIT Press.

Grossberg, S, & Schmajuk, NA (1989) Neural dynamics of adaptive timing and temporal discrimination during associative learning. *Neural Networks* **2**:79–102.

Haberly, LB (1990) Olfactory cortex. In GM Shepherd, ed., *The Synaptic Organization of the Brain*. New York: Oxford University Press.

Hahnloser, RH, Sarpeshkar, R, Mahowald, MA, Douglas, RJ, & Seung, HS (2000) Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature* **405**:947–951.

Hammer, M (1993) An identified neuron mediates the unconditioned stimulus in associative olfactory learning in honeybees. *Nature* **336**:59–63.

van Hateren, JH (1992) A theory of maximizing sensory information. *Biological Cybernetics* **68**:23–29.

van Hateren, JH (1993) Three modes of spatiotemporal preprocessing by eyes. *Journal of Comparative Physiology* **A172**:583–591.

Hebb, DO (1949) *The Organization of Behavior: A Neuropsychological Theory*. New York: Wiley.

of its frequency is called the power spectrum or power spectral density. White noise has a flat power spectrum.

White-Noise Stimuli

The defining characteristic of a white-noise stimulus is that its value at any one time is uncorrelated with its value at any other time. This condition can be expressed using the stimulus-stimulus correlation function, also called the stimulus autocorrelation, which is defined by analogy with equation 1.21 as

stimulus autocorrelation function Q_{ss}

$$Q_{ss}(\tau) = \frac{1}{T}\int_0^T dt\, s(t)s(t+\tau)\,. \tag{1.23}$$

Just as a correlation function provides information about the temporal relationship between two quantities, so an autocorrelation function tells us about how a quantity at one time is related to itself evaluated at another time. For white noise, the stimulus autocorrelation function is 0 in the range $-T/2 < \tau < T/2$ except when $\tau = 0$, and over this range

$$Q_{ss}(\tau) = \sigma_s^2\delta(\tau)\,. \tag{1.24}$$

The constant σ_s, which has the units of the stimulus times the square root of the unit of time, reflects the magnitude of the variability of the white noise. In appendix A, we show that equation 1.24 is equivalent to the statement that white noise has equal power at all frequencies.

No physical system can generate noise that is white to arbitrarily high frequencies. Approximations of white noise that are missing high-frequency components can be used, provided the missing frequencies are well above the sensitivity of the neuron under investigation. To approximate white noise, we consider times that are integer multiples of a basic unit of duration Δt, that is, times $t = m\Delta t$ for $m = 1, 2, \ldots, M$ where $M\Delta t = T$. The function $s(t)$ is then constructed as a discrete sequence of stimulus values. This produces a steplike stimulus waveform, like the one that appears in figure 1.8, with a constant stimulus value s_m presented during time bin m. In terms of the discrete-time values s_m, the condition that the stimulus is uncorrelated is

$$\frac{1}{M}\sum_{m=1}^{M} s_m s_{m+p} = \begin{cases} \sigma_s^2/\Delta t & \text{if } p = 0 \\ 0 & \text{otherwise}\,. \end{cases} \tag{1.25}$$

The factor of $1/\Delta t$ on the right side of this equation reproduces the δ function of equation 1.24 in the limit $\Delta t \to 0$. For approximate white noise, the autocorrelation function is 0 except for a region around $\tau = 0$ with width of order Δt. Similarly, the binning of time into discrete intervals of size Δt means that the noise generated has a flat power spectrum only up to frequencies of order $1/(2\Delta t)$.

DeAngelis, GC, Ohzawa, I, & Freeman, RD (1995) Receptive field dynamics in the central visual pathways. *Trends in Neuroscience* **18**:451–458.

Dempster, AP, Laird, NM, & Rubin, DB (1977) Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society* **B39**:1–38.

Destexhe, A, Mainen, Z, & Sejnowski, T (1994) Synthesis of models for excitable membranes, synaptic transmission and neuromodulation using a common kinetic formalism. *Journal of Computational Neuroscience* **1**:195–230.

De Valois, RL, & De Valois, KK (1990) *Spatial Vision*. New York: Oxford University Press.

Dickinson, A (1980) *Contemporary Animal Learning Theory*. Cambridge: Cambridge University Press.

Dong, DW, & Atick, JJ (1995) Temporal decorrelation: A theory of lagged and nonlagged responses in the lateral geniculate nucleus. *Network: Computation in Neural Systems* **6**:159–178.

Douglas, RJ, Koch, C, Mahowald, M, Martin, KAC, & Suarez, HH (1995) Recurrent excitation in neocortical circuits. *Science* **269**:981–985.

Douglas, RJ, & Martin, KAC (1998). Neocortex. In GM Shepherd, ed., *The Synaptic Organisation of the Brain. 4th ed.*, 459–509. Oxford: Oxford University Press.

Dowling, JE (1987) *The Retina: An Approachable Part of the Brain*. Cambridge, MA: Bellknap Press.

Dowling, JE (1992) *An Introduction to Neuroscience*. Cambridge, MA: Bellknap Press.

Duda, RO, & Hart, PE (1973) *Pattern Classification and Scene Analysis*. New York: Wiley.

Duda, RO, Hart, PE, & Stork, DG (2000) *Pattern Classification*. New York: Wiley.

Durbin, R, & Mitchison, G (1990) A dimension reduction framework for cortical maps. *Nature* **343**:644–647.

Durbin, R, & Willshaw, DJ (1987) An analogue approach to the travelling salesman problem using an elastic net method. *Nature* **326**:689–691.

Edelstein-Keshet, L (1988) *Mathematical Models in Biology*. New York: Random House.

Engel, AK, Konig, P, & Singer, W (1991) Direct physiological evidence for scene segmentation by temporal coding. *Proceedings of the National Academy of Sciences of the United States of America* **88**:9136–9140.

Enroth-Cugell, C, & Robson, JG (1966) The contrast sensitivity of retinal ganglion cells of the cat. *Journal of Physiology* **187**:517–522.

Ermentrout, GB (1998) Neural networks as spatio-temporal pattern-forming systems. *Reports on Progress in Physics* **64**:353–430.

Ermentrout, GB, & Cowan, J (1979) A mathematical theory of visual hallucination patterns. *Biological Cybernetics* **34**:137–150.

Erwin, E, Obermayer, K, & Schulten, K (1995) Models of orientation and ocular dominance columns in the visual cortex: A critical comparison. *Neural Computation* **7**:425–468.

Everitt, BS (1984) *An Introduction to Latent Variable Models*. London: Chapman & Hall.

Feller, W (1968) *An Introduction to Probability Theory and Its Application*. New York: Wiley.

average stimulus triggered on two spikes separated by 5 ± 1 ms. The average stimulus triggered on a pair of spikes separated by 5 ms is not the same as the sum of the average stimuli for each spike separately.

Spike-triggered averages of other stimulus-dependent quantities can provide additional insight into neural encoding, for example, spike-triggered average autocorrelation functions. Obviously, spike-triggered averages of higher-order stimulus combinations can be considered as well.

1.4 Spike-Train Statistics

A complete description of the stochastic relationship between a stimulus and a response would require us to know the probabilities corresponding to every sequence of spikes that can be evoked by the stimulus. Spike times are continuous variables, and, as a result, the probability for a spike to occur at any precisely specified time is actually zero. To get a nonzero value, we must ask for the probability that a spike occurs within a specified interval, for example, the interval between times t and $t + \Delta t$. For small Δt, the probability of a spike falling in this interval is proportional to the size of the interval, Δt. A similar relation holds for any continuous stochastic variable z. The probability that z takes a value between z and $z + \Delta z$, for small Δz (strictly speaking, as $\Delta z \to 0$), is equal to $p[z]\Delta z$, where $p[z]$ is called a probability density.

Throughout this book, we use the notation $P[\,]$ to denote probabilities and $p[\,]$ to denote probability densities. We use the bracket notation $P[\,]$ generically for the probability of something occurring and also to denote a specific probability function. In the latter case, the notation $P(\,)$ would be more appropriate, but switching between square brackets and parentheses is confusing, so the reader will have to use the context to distinguish between these cases.

The probability of a spike sequence appearing is proportional to the probability density of spike times, $p[t_1, t_2, \ldots, t_n]$. In other words, the probability $P[t_1, t_2, \ldots, t_n]$ that a sequence of n spikes occurs with spike i falling between times t_i and $t_i + \Delta t$ for $i = 1, 2, \ldots, n$ is given in terms of this density by the relation $P[t_1, t_2, \ldots, t_n] = p[t_1, t_2, \ldots, t_n](\Delta t)^n$.

Unfortunately, the number of possible spike sequences is typically so large that determining or even roughly estimating all of their probabilities of occurrence is impossible. Instead, we must rely on some statistical model that allows us to estimate the probability of an arbitrary spike sequence occurring, given our knowledge of the responses actually recorded. The firing rate $\mathsf{r}(t)$ determines the probability of firing a single spike in a small interval around the time t, but $\mathsf{r}(t)$ is not, in general, sufficient information to predict the probabilities of spike sequences. For example, the probability of two spikes occurring together in a sequence is not necessarily equal to the product of the probabilities that they occur individually, because

Berry, MJ, & Meister, M (1998) Refractoriness and neural precision. *Journal of Neuroscience* **18**: 2200–2211.

Bertsekas, DP, & Tsitsiklis, JN (1996) *Neuro-Dynamic Programming*. Belmont, MA: Athena Scientific.

Bialek, W, DeWeese, M, Rieke, F, & Warland, D (1993) Bits and brains: Information flow in the nervous system. *Physica* **A200**:581–593.

Bialek W, Rieke F, de Ruyter van Steveninck RR, & Warland D (1991) Reading a neural code. *Science* **252**:1854–1857.

Bienenstock, EL, Cooper, LN, & Munro, PW (1982) Theory for the development of neuron selectivity: Orientation specificity and binocular interaction in visual cortex. *Journal of Neuroscience* **2**:32–48.

Bishop, CM (1995) *Neural Networks for Pattern Recognition*. Oxford: Oxford University Press.

Blum, KI, & Abbott, LF (1996) A model of spatial map formation in the hippocampus of the rat. *Neural Computation* **8**:85–93.

Boas, ML (1966) *Mathematical Methods in the Physical Sciences*. New York: Wiley.

de Boer, E, & Kuyper, P (1968) Triggered correlation. *IEEE Biomedical Engineering* **15**:169–179.

Bower, JM, & Beeman, D (1998) *The Book of GENESIS: Exploring Realistic Neural Models with the GEneral NEural SImulation System*. Santa Clara, CA: Telos.

Braitenberg, V, & Schuz, A (1991) *Anatomy of the Cortex*. Berlin: Springer-Verlag.

Bressloff, PC, & Coombes, S (2000) Dynamics of strongly coupled spiking neurons. *Neural Computation* **12**:91–129.

Britten, KH, Shadlen, MN, Newsome, WT, & Movshon, JA (1992) The analysis of visual motion: A comparison of neuronal and psychophysical performance. *Journal of Neuroscience* **12**:4745–4765.

Brotchie PR, Andersen RA, Snyder LH, & Goodman SJ (1995) Head position signals used by parietal neurons to encode locations of visual stimuli. *Nature* **375**:232–235.

Burt, PJ, & Adelson, EH (1983) The Laplacian pyramid as a compact image code. *IEEE Transactions on Communications* **31**:532–540.

Bussgang, JJ (1952) Cross-correlation functions of amplitude-distorted Gaussian signals. *MIT Research Laboratory for Electronic Technology Report* **216**:1–14.

Bussgang, JJ (1975) Cross-correlation functions of amplitude-distorted Gaussian inputs. In AH Haddad, ed., *Nonlinear Systems*. Stroudsburg, PA: Dowden, Hutchinson & Ross.

Cajal, S Ramón y (1911) *Histologie du Système Nerveux de l'Homme et des Vertébrés*. (Translated by L Azoulay). Paris: Maloine. English translation by N Swanson, & LW Swanson (1995) *Histology of the Nervous Systems of Man and Vertebrates*. New York: Oxford University Press.

Campbell, FW, & Gubisch, RW (1966) Optical quality of the human eye. *Journal of Physiology* **186**:558–578.

Carandini M, Heeger DJ, & Movshon JA (1996) Linearity and gain control in V1 simple cells. In EG Jones, & PS Ulinski, eds. *Cerebral Cortex. Vol. 10, Cortical Models*. New York: Plenum Press.

Carandini, M, & Ringach, DL (1997). Predictions of a recurrent model of orientation selectivity. *Vision Research* **37**:3061–3071.

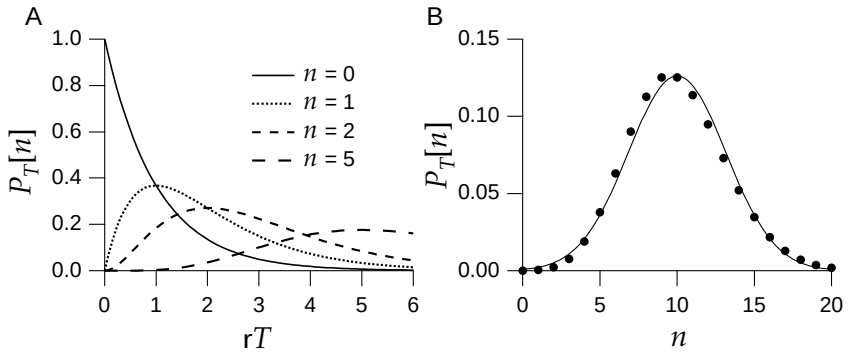

Figure 1.11 (A) The probability that a homogeneous Poisson process generates n spikes in a time period of duration T plotted for n = 0, 1, 2, and 5. The probability is plotted as function of the rate times the duration of the interval, rT, to make the plot applicable for any rate. (B) The probability of finding n spikes during a time period for which $rT = 10$ (dots) compared with a Gaussian distribution with mean and variance equal to 10 (line).

binomial coefficient $M!/(M-n)!n!$. Putting all these factors together, we find

$$P_T[n] = \lim_{\Delta t \to 0} \frac{M!}{(M-n)!n!} (r\Delta t)^n (1 - r\Delta t)^{M-n} . \tag{1.27}$$

To take the limit, we note that as $\Delta t \to 0$, M grows without bound because $M\Delta t = T$. Because n is fixed, we can write $M - n \approx M = T/\Delta t$. Using this approximation and defining $\epsilon = -r\Delta t$, we find that

$$\lim_{\Delta t \to 0} (1 - r\Delta t)^{M-n} = \lim_{\epsilon \to 0} \left((1+\epsilon)^{1/\epsilon}\right)^{-rT} = e^{-rT} = \exp(-rT) \tag{1.28}$$

because $\lim_{\epsilon \to 0}(1+\epsilon)^{1/\epsilon}$ is, by definition, $e = \exp(1)$. For large M, $M!/(M-n)! \approx M^n = (T/\Delta t)^n$, so

$$P_T[n] = \frac{(rT)^n}{n!} \exp(-rT) . \tag{1.29}$$

Poisson distribution

This is called the Poisson distribution. The probabilities $P_T[n]$, for a few n values, are plotted as a function of rT in figure 1.11A. Note that as n increases, the probability reaches its maximum at larger T values and that large n values are more likely than small ones for large T. Figure 1.11B shows the probabilities of various numbers of spikes occurring when the average number of spikes is 10. For large rT, which corresponds to a large expected number of spikes, the Poisson distribution approaches a Gaussian distribution with mean and variance equal to rT. Figure 1.11B shows that this approximation is already quite good for rT = 10.

We can compute the variance of spike counts produced by a Poisson process from the probabilities in equation 1.29. For spikes counted over an interval of duration T, the variance of the spike count (derived in appendix B) is

$$\sigma_n^2 = \langle n^2 \rangle - \langle n \rangle^2 = rT . \tag{1.30}$$

References

Abbott, LF (1992) Simple diagrammatic rules for solving dendritic cable problems. *Physica* **A185**:343–356.

Abbott, LF (1994) Decoding neuronal firing and modeling neural networks. *Quarterly Review of Biophysics* **27**:291–331.

Abbott, LF, & Blum, KI (1996) Functional significance of long-term potentiation for sequence learning and prediction. *Cerebral Cortex* **6**:406–416.

Abbott, LF, Farhi, E, & Gutmann, S (1991) The path integral for dendritic trees. *Biological Cybernetics* **66**:49–60.

Abbott, LF, Varela, JA, Sen, K, & Nelson, SB (1997) Synaptic depression and cortical gain control. *Science* **275**:220–224.

Adelson, EH, & Bergen, JR (1985) Spatiotemporal energy models for the perception of motion. *Journal of the Optical Society of America* **A2**:284–299.

Ahmed, B, Anderson, JC, Douglas, RJ, Martin, KAC, & Whitterage, D (1998) Estimates of the net excitatory currents evoked by visual stimulation of identified neurons in cat visual cortex. *Cerebral Cortex* **8**:462–476.

Amari, S (1999) Natural gradient learning for over- and under-complete bases in ICA. *Neural Computation* **11**:1875–1883.

Amit, DJ (1989) *Modeling Brain Function*. New York: Cambridge University Press.

Amit, DJ, & Tsodyks, MV (1991a) Quantitative study of attractor neural networks retrieving at low spike rates. I. Substrate-spikes, rates and neuronal gain. *Network: Computation in Neural Systems* **2**:259–273.

Amit, DJ, & Tsodyks, MV (1991b) Quantitative study of attractor neural networks retrieving at low spike rates. II. Low-rate retrieval in symmetric networks. *Network: Computation in Neural Systems* **2**:275–294.

Andersen, RA (1989) Visual and eye movement functions of posterior parietal cortex. *Annual Review of Neuroscience* **12**:377–403.

Atick, JJ (1992) Could information theory provide an ecological theory of sensory processing? *Network: Computation in Neural Systems* **3**:213–251.

Atick, JJ, Li, Z, & Redlich, AN (1992) Understanding retinal color coding from first principles. *Neural Computation* **4**:559–572.

Atick, JJ, & Redlich, AN (1990) Towards a theory of early visual processing. *Neural Computation* **2**:308–320.

Atick, JJ, & Redlich, AN (1993) Convergent algorithm for sensory receptive field development. *Neural Computation* **5**:45–60.

Baddeley, R, Abbott, LF, Booth, MJA, Sengpiel, F, Freeman, T, Wakeman, EA, & Rolls, ET (1997) Responses of neurons in primary and interior temporal visual cortices to natural scenes. *Proceedings of the Royal Society of London*. **B264**:1775–1783.

a train. This is called the spike-train autocorrelation function, and it is particularly useful for detecting patterns in spike trains, most notably oscillations. The spike-train autocorrelation function is the autocorrelation of the neural response function of equation 1.1 with its average over time and trials subtracted out. The time average of the neural response function, from equation 1.4, is the spike-count rate r, and the trial average of this quantity is $\langle r \rangle = \langle n \rangle / T$. Thus, the spike-train autocorrelation function is

spike-train autocorrelation function $Q_{\rho\rho}$

$$Q_{\rho\rho}(\tau) = \frac{1}{T}\int_0^T dt\, \langle (\rho(t) - \langle r \rangle)\,(\rho(t+\tau) - \langle r \rangle)\rangle \,. \tag{1.35}$$

Because the average is subtracted from the neural response function in this expression, $Q_{\rho\rho}$ should really be called an autocovariance, not an autocorrelation, but in practice it isn't.

The spike-train autocorrelation function is constructed from data in the form of a histogram by dividing time into bins. The value of the histogram for a bin labeled with a positive or negative integer m is computed by determining the number of the times that any two spikes in the train are separated by a time interval lying between $(m - 1/2)\,\Delta t$ and $(m + 1/2)\,\Delta t$ with Δt the bin size. This includes all pairings, even between a spike and itself. We call this number N_m. If the intervals between the n^2 spike pairs in the train were uniformly distributed over the range from 0 to T, there would be $n^2\,\Delta t / T$ intervals in each bin. This uniform term is removed from the autocorrelation histogram by subtracting $n^2\,\Delta t / T$ from N_m for all m. The spike-train autocorrelation histogram is then defined by dividing the resulting numbers by T, so the value of the histogram in bin m is $H_m = N_m/T - n^2\,\Delta t / T^2$. For small bin sizes, the $m = 0$ term in the histogram counts the average number of spikes, that is $N_0 = \langle n \rangle$ and in the limit $\Delta t \to 0$, $H_0 = \langle n \rangle / T$ is the average firing rate $\langle r \rangle$. Because other bins have H_m of order Δt, the large $m = 0$ term is often removed from histogram plots. The spike-train autocorrelation function is defined as $H_m / \Delta t$ in the limit $\Delta t \to 0$, and it has the units of a firing rate squared. In this limit, the $m = 0$ bin becomes a δ function, $H_0 / \Delta t \to \langle r \rangle \delta(\tau)$.

As we have seen, the distribution of interspike intervals for adjacent spikes in a homogeneous Poisson spike train is exponential (equation 1.31). By contrast, the intervals between any two spikes (not necessarily adjacent) in such a train are uniformly distributed. As a result, the subtraction procedure outlined above gives $H_m = 0$ for all bins except for the $m = 0$ bin that contains the contribution of the zero intervals between spikes and themselves. The autocorrelation function for a Poisson spike train generated at a constant rate $\langle r \rangle = \mathsf{r}$ is thus

$$Q_{\rho\rho}(\tau) = \mathsf{r}\,\delta(\tau) \,. \tag{1.36}$$

cross-correlation function

A cross-correlation function between spike trains from two different neurons can be defined by analogy with the autocorrelation function by determining the distribution of intervals between pairs of spikes, one taken

In addition to finite, sample spaces can be either countably or uncountably infinite. In the latter case, there are various technical complications that are discussed in the references. Under suitable conditions, a continuous random variable S, which is a mapping from a sample space to a continuous space such as the real numbers, has a probability density function $p[s]$ defined by

continuous random variable

probability density

$$p[s] = \lim_{\Delta s \to 0} \left(\frac{P[s \leq S \leq s + \Delta s]}{\Delta s} \right) . \tag{A.86}$$

Quantities such as the mean and variance of a continuous random variable are defined as for a discrete random variable, but involve integrals over probability densities rather than sums over probabilities.

Some commonly used discrete and continuous distributions are listed in the table below.

Name	Range of s	Probability	Mean	Variance
Bernoulli	$s = 0$ or 1	$p^s(1-p)^{1-s}$	p	$p(1-p)$
Poisson	$s = 0, 1, 2, \ldots$	$\alpha^s \exp(-\alpha)/s!$	α	α
Exponential	$s > 0$	$\alpha \exp(-\alpha s)$	$1/\alpha$	$1/\alpha^2$
Gaussian	$-\infty < s < \infty$	$\mathcal{N}[s; g, \Sigma]$	g	Σ
Cauchy	$-\infty < s < \infty$	$\beta/(\pi((s-\alpha)^2 + \beta^2))$	$*\,\alpha\,*$	$*\,1/\beta^2\,*$

where

$$\mathcal{N}(s; g, \Sigma) = \frac{1}{\sqrt{2\pi\Sigma}} \exp\left(-\frac{(s-g)^2}{2\Sigma} \right) . \tag{A.87}$$

Here, we use Σ to denote the variance of the Gaussian distribution, which is more often written as σ^2. The asterisks in the entries for the Cauchy distribution reflect the fact that it has such heavy tails that the integrals defining its mean and variance do not converge. Nevertheless, α and $1/\beta^2$ play similar roles, and are called location and scale parameters respectively.

The Gaussian distribution is particularly important because of the central limit theorem. Consider m continuous random variables $S_1, S_2, S_3, \ldots S_m$ that are independent and have identical distributions with finite mean g and variance σ^2. Defining

central limit theorem

$$Z_m = \frac{1}{m} \sum_{k=1}^{m} S_k , \tag{A.88}$$

the central limit theorem states that, under rather general conditions,

$$\lim_{m \to \infty} P\left[\frac{\sqrt{m}(Z_m - g)}{\sigma} \leq s \right] = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{s} dz \, \exp(-z^2/2) \tag{A.89}$$

for every s. This implies that for large m, Z_m should be approximately Gaussian distributed with mean g and variance σ^2/m.

This result applies if the spike times have been written in temporal order. If the spike times are not ordered, so that, for example, we are interested in the probability density for any spike occurring at the time t_1, not necessarily the first spike, this expression should be divided by a factor of $n!$ to account for the number of different possible orderings of spike times.

The Poisson Spike Generator

Spike sequences can be simulated by using some estimate of the firing rate, $r_{est}(t)$, predicted from knowledge of the stimulus, to drive a Poisson process. A simple procedure for generating spikes in a computer program is based on the fact that the estimated probability of firing a spike during a short interval of duration Δt is $r_{est}(t)\Delta t$. The program progresses through time in small steps of size Δt and generates, at each time step, a random number x_{rand} chosen uniformly in the range between 0 and 1. If $r_{est}(t)\Delta t > x_{rand}$ at that time step, a spike is fired; otherwise it is not.

For a constant firing rate, it is faster to compute spike times t_i for $i = 1, 2, \ldots n$ iteratively by generating interspike intervals from an exponential probability density (equation 1.31). If x_{rand} is uniformly distributed over the range between 0 and 1, the negative of its logarithm is exponentially distributed. Thus, we can generate spike times iteratively from the formula $t_{i+1} = t_i - \ln(x_{rand})/r$. Unlike the algorithm discussed in the previous paragraph, this method works only for constant firing rates. However, it can be extended to time-dependent rates by using a procedure called rejection sampling or spike thinning. The thinning technique requires a bound r_{max} on the estimated firing rate such that $r_{est}(t) \leq r_{max}$ at all times. We first generate a spike sequence corresponding to the constant rate r_{max} by iterating the rule $t_{i+1} = t_i - \ln(x_{rand})/r_{max}$. The spikes are then thinned by generating another x_{rand} for each i and removing the spike at time t_i from the train if $r_{est}(t_i)/r_{max} < x_{rand}$. If $r_{est}(t_i)/r_{max} \geq x_{rand}$, spike i is retained. Thinning corrects for the difference between the estimated time-dependent rate and the maximum rate.

Figure 1.13 shows an example of a model of an orientation-selective V1 neuron constructed in this way. In this model, the estimated firing rate is determined from the response tuning curve of figure 1.5B,

$$r_{est}(t) = f(s(t)) = r_{max} \exp\left(-\frac{1}{2}\left(\frac{s(t) - s_{max}}{\sigma_f}\right)^2\right). \tag{1.38}$$

This is an extremely simplified model of response dynamics, because the firing rate at any given time depends only on the value of the stimulus at that instant of time and not on its recent history. Models that allow for a dependence of firing rate on stimulus history are discussed in chapter 2. In figure 1.13, the orientation angle increases in a sequence of steps. The firing rate follows these changes, and the Poisson process generates an irregular firing pattern that reflects the underlying rate but varies from trial to trial.

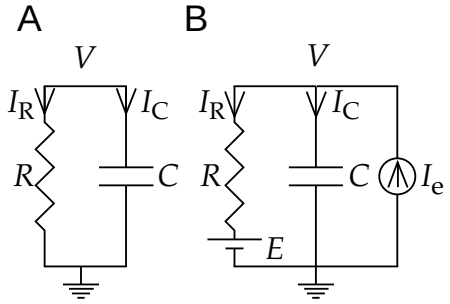

Figure A.2 RC circuits. (A) Current $I_C = -I_R$ flows in the resistor-capacitor circuit as the stored charge is released. (B) Simple passive membrane model including a potential E and current source I_e. As in figure A.1, the lined triangles represent a ground or point of 0 voltage.

showing the exponential decay (with time constant $\tau = RC$) of the initial voltage $V(0)$ as the charge on the capacitor leaks out through the resistor.

Figure A.2B includes two extra components needed to build a simple model neuron, the voltage source E and the current source I_e. Using Kirchhoff's laws, $I_e - I_C - I_R = 0$, and the equation for the voltage V is

$$C\frac{dV}{dt} = \frac{E - V}{R} + I_e\,. \tag{A.79}$$

If I_e is constant, the solution of this equation is

$$V(t) = V_\infty + (V(0) - V_\infty)\exp(-t/\tau)\,, \tag{A.80}$$

where $V_\infty = E + RI_e$ and $\tau = RC$. This shows exponential relaxation from the initial potential $V(0)$ to the equilibrium potential V_∞ at a rate governed by the time constant τ of the circuit.

For the case $I_e = I\cos(\omega t)$, once an initial transient has decayed to 0, we find

$$V(t) = E + \frac{RI\cos(\omega t - \phi)}{\sqrt{1 + \omega^2\tau^2}}\,, \tag{A.81}$$

where $\tan(\phi) = \omega\tau$. Equation A.81 shows that the cell membrane acts as a low-pass filter, because the higher the frequency ω of the input current, the greater the attenuation of the oscillations of the potential due to the factor $1/(1 + \omega^2\tau^2)^{1/2}$. The phase shift ϕ is an increasing function of frequency that approaches $\pi/2$ as $\omega \to \infty$.

A.5 Probability Theory

Probability distributions and densities are discussed extensively in the text. Here, we present a slightly more formal treatment. At the heart of

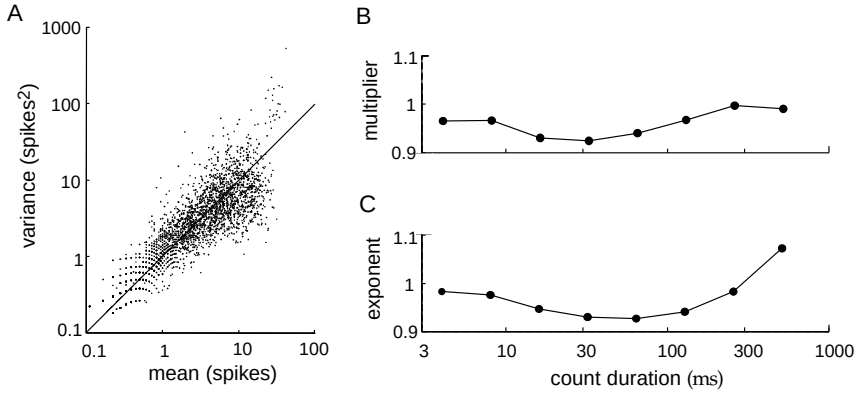

Figure 1.14 Variability of MT neurons in alert macaque monkeys responding to moving visual images. (A) Variance of the spike counts for a 256 ms counting period plotted against the mean spike count. The straight line is the prediction of the Poisson model. Data are from 94 cells recorded under a variety of stimulus conditions. (B) The multiplier A in the relationship between spike-count variance and mean as a function of the duration of the counting interval. (C) The exponent B in this relation as a function of the duration of the counting interval. (Adapted from O'Keefe et al., 1997.)

The Fano factor describes the relationship between the mean spike count over a given interval and the spike-count variance. Mean spike counts $\langle n \rangle$ and variances σ_n^2 from a wide variety of neuronal recordings have been fitted to the equation $\sigma_n^2 = A\langle n \rangle^B$, and the multiplier A and exponent B have been determined. The values of both A and B typically lie between 1.0 and 1.5. Because the Poisson model predicts $A = B = 1$, this indicates that the data show a higher degree of variability than the Poisson model would predict. However, many of these experiments involve anesthetized animals, and it is known that response variability is higher in anesthetized than in alert animals.

area MT

Figure 1.14 shows data for spike-count means and variances extracted from recordings of MT neurons in alert macaque monkeys using a number of different stimuli. The MT (medial temporal) area is a visual region of the primate cortex where many neurons are sensitive to image motion. The individual means and variances are scattered in figure 1.14A, but they cluster around the diagonal which is the Poisson prediction. Similarly, the results show A and B values close to 1, the Poisson values (figure 1.14B). Of course, many neural responses cannot be described by Poisson statistics, but it is reassuring to see a case where the Poisson model seems a reasonable approximation. As mentioned previously, when spike trains are not described very accurately by a Poisson model, refractory effects are often the primary reason.

Interspike interval distributions are extracted from data as interspike interval histograms by counting the number of intervals falling in discrete time bins. Figure 1.15A presents an example from the responses of a nonbursting cell in area MT of a monkey in response to images consisting of

differential equation. Linearization about equilibrium points can be used to analyze nonlinear difference equations as well as differential equations, and this reveals similar classes of behavior. We illustrate difference equations by analyzing a linear case, *difference equation*

$$\mathbf{v}(n+1) = \mathbf{v}(n) + \mathbf{W} \cdot \mathbf{v}(n) \,. \tag{A.71}$$

The strategy for solving this equation is similar to that for solving differential equations. Assuming $\mathbf{W}$ has a complete set of linearly independent eigenvectors $\mathbf{e}_1, \ldots, \mathbf{e}_N$ with different eigenvalues $\lambda_1, \ldots, \lambda_N$, the modes separate, and the general solution is

$$\mathbf{v}(n) = \sum_{\mu=1}^{N} c_\mu (1+\lambda_\mu)^n \mathbf{e}_\mu \,, \tag{A.72}$$

where $\mathbf{v}(0) = \sum_\mu c_\mu \mathbf{e}_\mu$. This has characteristics similar to equation A.66. Writing $\lambda_\mu = \alpha_\mu + i\omega_\mu$, mode μ is oscillatory if $\omega_\mu \neq 0$. In the discrete case, stability of the system is controlled by the magnitude

$$|1+\lambda_\mu|^2 = \left(1+\alpha_\mu\right)^2 + \left(\omega_\mu\right)^2 . \tag{A.73}$$

If this is greater than 1 for any value of μ, $|\mathbf{v}(n)| \to \infty$ as $n \to \infty$. If it is less than 1 for all μ, $\mathbf{v}(n) \to \mathbf{0}$ in this limit.

A.4 Electrical Circuits

Biophysical models of single cells involve equivalent circuits composed of resistors, capacitors, and voltage and current sources. We review here basic results for such circuits. Figures A.1A and A.1B show the standard symbols for resistors and capacitors, and define the relevant voltages and currents. A resistor (figure A.1A) satisfies Ohm's law, which states that the voltage $V_\mathrm{R} = V_1 - V_2$ across a resistance R carrying a current I_R is *Ohm's law*

$$V_\mathrm{R} = I_\mathrm{R} R \,. \tag{A.74}$$

Resistance is measured in ohms (Ω); 1 ohm is the resistance through which 1 ampere of current causes a voltage drop of 1 volt (1 V = 1 A $\times$ 1 Ω).

A capacitor (figure A.1B) stores charge across an insulating medium, and the voltage across it $V_\mathrm{C} = V_1 - V_2$ is related to the charge it stores, Q_C, by

$$CV_\mathrm{C} = Q_\mathrm{C} \,, \tag{A.75}$$

where C is the capacitance. Electrical current cannot cross the insulating medium, but charges can be redistributed on each side of the capacitor, which leads to the flow of current. We can take a time derivative of both sides of equation A.75 and use the fact that current is equal to the rate of

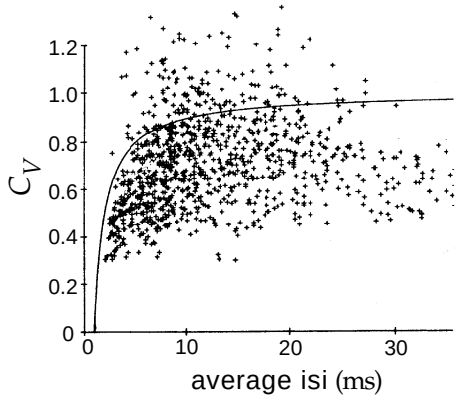

Figure 1.16 Coefficients of variation for a large number of V1 and MT neurons plotted as a function of mean interspike interval. The solid curve is the result of a Poisson model with a refractory period. (Adapted from Softky and Koch, 1992.)

plicity. However, there are cases in which the accuracy in the timing and numbers of spikes fired by a neuron is considerably higher than would be implied by Poisson statistics. Furthermore, even when it successfully describes data, the Poisson model does not provide a mechanistic explanation of neuronal response variability. Spike generation, by itself, is highly reliable in real neurons. Figure 1.17 compares the response of V1 cells to constant current injection in vivo and in vitro. The in vitro response is a regular and reproducible spike train (left panel). The same current injection paradigm applied in vivo produces a highly irregular pattern of firing (center panel) similar to the response to a moving bar stimulus (right panel). Although some of the basic statistical properties of firing variability may be captured by the Poisson model of spike generation, the spike-generating mechanism itself in real neurons is clearly not responsible for the variability. We explore ideas about possible sources of spike-train variability in chapter 5.

Some neurons fire action potentials in clusters or bursts of spikes that cannot be described by a Poisson process with a fixed rate. Bursting can be included in a Poisson model by allowing the firing rate to fluctuate in order to describe the high rate of firing during a burst. Sometimes the distribution of bursts themselves can be described by a Poisson process (such a doubly stochastic process is called a Cox process).

1.5 The Neural Code

The nature of the neural code is a topic of intense debate within the neuroscience community. Much of the discussion has focused on whether neurons use rate coding or temporal coding, often without a clear definition of what these terms mean. We feel that the central issue in neural coding is whether individual action potentials and individual neurons encode inde-

of equation A.59, we must have $\mathbf{f}(\mathbf{v}_\infty) = 0$. General solutions of equation A.59 when $\mathbf{f}$ is nonlinear cannot be constructed, but we can use linear techniques to study the behavior of $\mathbf{v}$ near a fixed point $\mathbf{v}_\infty$. If $\mathbf{f}$ is linear, the techniques we use and the solutions we obtain as approximations in the nonlinear case are exact. Near the fixed point $\mathbf{v}_\infty$, we write

$$\mathbf{v}(t) = \mathbf{v}_\infty + \boldsymbol{\epsilon}(t) \tag{A.60}$$

and consider the case when all the components of the vector $\boldsymbol{\epsilon}$ are small. Then, we can expand $\mathbf{f}$ in a Taylor series, *Taylor series*

$$\mathbf{f}(\mathbf{v}(t)) \approx \mathbf{f}(\mathbf{v}_\infty) + \mathbf{J} \cdot \boldsymbol{\epsilon}(t) = \mathbf{J} \cdot \boldsymbol{\epsilon}(t)\,, \tag{A.61}$$

where $\mathbf{J}$ is the called the Jacobian matrix and has elements *Jacobian matrix*

$$J_{ab} = \left.\frac{\partial f_a(\mathbf{v})}{\partial v_b}\right|_{\mathbf{v}=\mathbf{v}_\infty}\,. \tag{A.62}$$

In the second equality of equation A.61, we have used the fact that $\mathbf{f}(\mathbf{v}_\infty) = 0$.

Using the approximation of equation A.61, equation A.59 becomes

$$\frac{d\boldsymbol{\epsilon}}{dt} = \mathbf{J} \cdot \boldsymbol{\epsilon}\,. \tag{A.63}$$

The temporal evolution of $\mathbf{v}(t)$ is best understood by expanding $\boldsymbol{\epsilon}$ in the basis provided by the eigenvectors of $\mathbf{J}$. Assuming that $\mathbf{J}$ is real and has N linearly independent eigenvectors $\mathbf{e}_1, \ldots, \mathbf{e}_N$ with different eigenvalues $\lambda_1, \ldots, \lambda_N$, we write

$$\boldsymbol{\epsilon}(t) = \sum_{\mu=1}^{N} c_\mu(t)\mathbf{e}_\mu\,. \tag{A.64}$$

Substituting this into equation A.63, we find that the coefficients must satisfy

$$\frac{dc_\mu}{dt} = \lambda_\mu c_\mu\,. \tag{A.65}$$

This produces the solution

$$\boldsymbol{\epsilon}(t) = \sum_{\mu=1}^{N} c_\mu(0)\exp(\lambda_\mu t)\mathbf{e}_\mu\,, \tag{A.66}$$

where $\boldsymbol{\epsilon}(0) = \sum_\mu c_\mu(0)\mathbf{e}_\mu$. The individual terms in the sum on the right side of equation A.66 are called modes. This solution is exact for equation A.63, but is only a valid approximation when applied to equation A.59 if $\boldsymbol{\epsilon}$ is small. Note that the different coefficients c_μ evolve over time, independently of each other. This does not require the eigenvectors to be orthogonal. If the eigenvalues and eigenvectors are complex, $\mathbf{v}(t)$ will nonetheless remain real if $\mathbf{v}(0)$ is real, because the complex modes come *modes*

Independent-spike codes are much simpler to analyze than correlation codes, and most work on neural coding assumes spike independence. When careful studies have been done, it has been found that some information is carried by correlations between two or more spikes, but this information is rarely larger than 10% of the information carried by spikes considered independently. Of course, it is possible that, due to our ignorance of the "real" neural code, we have not yet uncovered or examined the types of correlations that are most significant for neural coding. Although this is not impossible, we view it as unlikely and feel that the evidence for independent-spike coding, at least as a fairly accurate approximation, is quite convincing.

The discussion to this point has focused on information carried by single neurons, but information is typically encoded by neuronal populations. When we study population coding, we must consider whether individual neurons act independently, or whether correlations between different neurons carry additional information. The analysis of population coding is easiest if the response of each neuron is considered statistically independent, and such independent-neuron coding is typically assumed in the analysis of population codes (chapter 3). The independent-neuron hypothesis does not mean that the spike trains of different neurons are not combined into an ensemble code. Rather, it means that they can be combined without taking correlations into account. To test the validity of this assumption, we must ask whether correlations between the spiking of different neurons provide additional information about a stimulus that cannot be obtained by considering all of their firing patterns individually.

independent-neuron code

synchrony and oscillations

Synchronous firing of two or more neurons is one mechanism for conveying information in a population correlation code. Rhythmic oscillations of population activity provide another possible mechanism, as discussed below. Both synchronous firing and oscillations are common features of the activity of neuronal populations. However, the existence of these features is not sufficient for establishing a correlation code, because it is essential to show that a significant amount of information is carried by the resulting correlations. The assumption of independent-neuron coding is a useful simplification that is not in gross contradiction with experimental data, but it is less well established and more likely to be challenged in the future than the independent-spike hypothesis.

hippocampal place cells

Place-cell coding of spatial location in the rat hippocampus is an example in which at least some additional information appears to be carried by correlations between the firing patterns of neurons in a population. The hippocampus is a structure located deep inside the temporal lobe that plays an important role in memory formation and is involved in a variety of spatial tasks. The firing rates of many hippocampal neurons, recorded when a rat is moving around a familiar environment, depend on the location of the animal and are restricted to spatially localized areas called the place fields of the cells. In addition, when a rat explores an environment, hippocampal neurons fire collectively in a rhythmic pattern with a frequency in the theta range, 7-12 Hz. The spiking time of an individual

The condition that characterizes an extreme value of the function $f(\mathbf{v})$ is that small changes $\Delta\mathbf{v}$ (with components Δv_a) in the vector $\mathbf{v}$ should not change the value of the function to first order in $\Delta\mathbf{v}$. This results in the condition

$$\sum_{a=1}^{N}[\nabla f]_a \Delta v_a = 0\,, \tag{A.50}$$

where we use the notation

$$[\nabla f]_a = \frac{\partial f}{\partial v_a} \tag{A.51}$$

to make the equations more compact. Without a constraint, equation A.50 must be satisfied for all $\Delta\mathbf{v}$, which can occur only if each term in the sum vanishes separately. Thus, we find the usual condition for an extremum

$$[\nabla f]_a = 0 \tag{A.52}$$

for all a. However, with a constraint such as $g(\mathbf{v})$=constant, equation A.50 does not have to hold for all possible $\Delta\mathbf{v}$, only for those that satisfy the constraint. The condition on $\Delta\mathbf{v}$ imposed by the constraint is that g cannot change to first order in $\Delta\mathbf{v}$. Therefore,

$$\sum_{a=1}^{N}[\nabla g]_a \Delta v_a = 0 \tag{A.53}$$

with the same notation for the derivative used for g as for f.

The most obvious way to deal with the constraint equation A.53 is to solve for one of the components of $\Delta\mathbf{v}$, say Δv_c, writing

$$\Delta v_c = -\frac{1}{[\nabla g]_c}\sum_{a\neq c}[\nabla g]_a \Delta v_a\,. \tag{A.54}$$

Then we substitute this expression into equation A.50 to obtain

$$\sum_{a\neq c}[\nabla f]_a \Delta v_a - \frac{[\nabla f]_c}{[\nabla g]_c}\sum_{a\neq c}[\nabla g]_a \Delta v_a = 0\,. \tag{A.55}$$

Because we have eliminated the constraint, this equation must be satisfied for all values of the remaining components of $\Delta\mathbf{v}$, those with $a \neq c$, and thus we find

$$[\nabla f]_a - \frac{[\nabla f]_c}{[\nabla g]_c}[\nabla g]_a = 0 \tag{A.56}$$

for all $a \neq c$. The derivatives of f and g are functions of $\mathbf{v}$, so these equations can be solved to determine where the extremum point is located.

In the above derivation, we have singled out component c for special treatment. We have no way of knowing until we get to the end of the calculation whether the particular c we chose leads to a simple or a complex set of

The temporal structure of a spike train or firing rate evoked by a stimulus is determined both by the dynamics of the stimulus and by the nature of the neural encoding process. Stimuli that change rapidly tend to generate precisely timed spikes and rapidly changing firing rates no matter what neural coding strategy is being used. Temporal coding refers to (or should refer to) temporal precision in the response that does not arise solely from the dynamics of the stimulus, but that nevertheless relates to properties of the stimulus. The interplay between stimulus and encoding dynamics makes the identification of a temporal code difficult.

The issue of temporal coding is distinct and independent from the issue of independent-spike coding discussed above. If the independent-spike hypothesis is valid, the temporal character of the neural code is determined by the behavior of $r(t)$. If $r(t)$ varies slowly with time, the code is typically called a rate code, and if it varies rapidly, the code is called temporal. Figure 1.19 provides an example of different firing-rate behaviors for a neuron in area MT of a monkey recorded over multiple trials with three different stimuli (consisting of moving random dots). The activity in the top panel would typically be regarded as reflecting rate coding, and the activity in the bottom panel as reflecting temporal coding. However, the identification of rate and temporal coding in this way is ambiguous because it is not obvious what criterion should be used to characterize the changes in $r(t)$ as slow or rapid.

One possibility is to use the spikes to distinguish slow from rapid, so that a temporal code is identified when peaks in the firing rate occur with roughly the same frequency as the spikes themselves. In this case, each peak corresponds to the firing of only one, or at most a few action potentials. While this definition makes intuitive sense, it is problematic to extend it to the case of population coding. When many neurons are involved, any single neuron may fire only a few spikes before its firing rate changes, but collectively the population may produce a large number of spikes over the same time period. Thus, by this definition, a neuron that appears to employ a temporal code may be part of a population that does not.

Another proposal is to use the stimulus, rather than the response, to establish what makes a temporal code. In this case, a temporal code is defined as one in which information is carried by details of spike timing on a scale shorter than the fastest time characterizing variations of the stimulus. This requires that information about the stimulus be carried by Fourier components of $r(t)$ at frequencies higher than those present in the stimulus. Many of the cases where a temporal code has been reported using spikes to define the nature of the code would be called rate codes if the stimulus were used instead.

The debate between rate and temporal coding dominates discussions about the nature of the neural code. Determining the temporal resolution of the neural code is clearly important, but much of this debate seems uninformative. We feel that the central challenge is to identify relationships

which is equivalent to equation A.39. A related result is Parseval's theorem, *Parseval's theorem*

$$\int_{-\infty}^{\infty} dt\, |f(t)|^2 = \frac{1}{2\pi} \int_{-\infty}^{\infty} d\omega\, |\tilde{f}(\omega)|^2 \,. \tag{A.41}$$

If $f(t)$ is periodic with period T (so that $f(t+T) = f(t)$ for all t), it can be represented by a Fourier series rather than a Fourier integral. That is, *periodic function* *Fourier series*

$$f(t) = \sum_{k=-\infty}^{\infty} \tilde{f}_k \exp(-i2\pi kt/T) \,, \tag{A.42}$$

where $\tilde{f}_k$ is given by

$$\tilde{f}_k = \frac{1}{T} \int_0^T dt\, f(t) \exp(i2\pi kt/T) \,. \tag{A.43}$$

As in the case of Fourier transforms, certain conditions have to hold for the series to converge and to be exactly invertible. The Fourier series has properties similar to Fourier transforms, including a convolution theorem and a version of Parseval's theorem. The real and imaginary parts of a Fourier series are often separated, giving the alternative form

$$f(t) = \tilde{f}_0 + \sum_{k=1}^{\infty} \left(\tilde{f}_k^{\mathrm{c}} \cos(2\pi kt/T) + \tilde{f}_k^{\mathrm{s}} \sin(2\pi kt/T) \right) \tag{A.44}$$

with

$$\begin{aligned} \tilde{f}_0 &= \frac{1}{T} \int_0^T dt\, f(t) \,, \quad \tilde{f}_k^{\mathrm{c}} = \frac{2}{T} \int_0^T dt\, f(t) \cos(2\pi kt/T) \,, \\ \tilde{f}_k^{\mathrm{s}} &= \frac{2}{T} \int_0^T dt\, f(t) \sin(2\pi kt/T) \,. \end{aligned} \tag{A.45}$$

When computed numerically, a Fourier transform is typically based on a certain number, N_t, of samples of the function, $f_n = f(n\delta)$ for $n = 0, 1, \ldots N_t - 1$. The discrete Fourier transform of these samples is then used as an approximation of the continuous Fourier transform. The discrete Fourier transform is defined as *discrete Fourier transform*

$$\tilde{f}_m = \sum_{n=0}^{N_t-1} f_n \exp\left(i2\pi nm/N_t\right) \,. \tag{A.46}$$

Note that $\tilde{f}_{N_t+m} = \tilde{f}_m$. An approximation of the continuous Fourier transform is provided by the relation $\tilde{f}(2\pi m/(N_t\delta)) \approx \delta \tilde{f}_m$. The inverse discrete Fourier transform is

$$f_n = \frac{1}{N_t} \sum_{m=0}^{N_t-1} \tilde{f}_m \exp\left(-i2\pi mn/N_t\right) \,. \tag{A.47}$$

and the average firing rate $\langle r \rangle$. In the discussion of how the firing rate $r(t)$ could be extracted from data, we introduced the important concepts of a linear filter and a kernel acting as a sliding window function. The average firing rate expressed as a function of a static stimulus parameter is called the response tuning curve, and we presented examples of Gaussian, cosine, and sigmoidal tuning curves. Spike-triggered averages of stimuli, or reverse correlation functions, were introduced to characterize the selectivity of neurons to dynamic stimuli. The homogeneous and inhomogeneous Poisson processes were presented as models of stochastic spike sequences. We defined correlation functions, auto- and cross-correlations, and power spectra, and used the Fano factor, interspike-interval histogram, and coefficient of variation to characterize the stochastic properties of spiking. We concluded with a discussion of independent-spike and independent-neuron codes versus correlation codes, and of the temporal precision of spike timing as addressed in discussions of temporal coding.

1.7 Appendices

A: The Power Spectrum of White Noise

The Fourier transform of the stimulus autocorrelation function (see the Mathematical Appendix),

$$\tilde{Q}_{ss}(\omega) = \frac{1}{T}\int_{-T/2}^{T/2} d\tau\, Q_{ss}(\tau)\exp(i\omega\tau)\,, \tag{1.40}$$

power spectrum

is called the power spectrum. Because we have defined the stimulus as periodic outside the range of the trial T, we have used a finite-time Fourier transform and ω should be restricted to values that are integer multiples of $2\pi/T$. We can compute the power spectrum for a white-noise stimulus using the fact that $Q_{ss}(\tau) = \sigma_s^2\delta(\tau)$ for white noise,

$$\tilde{Q}_{ss}(\omega) = \frac{\sigma_s^2}{T}\int_{-T/2}^{T/2} d\tau\,\delta(\tau)\exp(i\omega\tau) = \frac{\sigma_s^2}{T}\,. \tag{1.41}$$

This is the defining characteristic of white noise; its power spectrum is independent of frequency.

Using the definition of the stimulus autocorrelation function, we can also write

$$\begin{aligned}\tilde{Q}_{ss}(\omega) &= \frac{1}{T}\int_0^T dt\, s(t)\frac{1}{T}\int_{-T/2}^{T/2} d\tau\, s(t+\tau)\exp(i\omega\tau) \qquad (1.42)\\ &= \frac{1}{T}\int_0^T dt\, s(t)\exp(-i\omega t)\frac{1}{T}\int_{-T/2}^{T/2} d\tau\, s(t+\tau)\exp(i\omega(t+\tau))\,.\end{aligned}$$

The sequence of functions used to construct the δ function as a limit is not unique. In essence, any function that integrates to 1 and has a single peak that gets continually narrower and taller as the limit is taken can be used. For example, the δ function can be expressed as the limit of a square pulse

$$\delta(t) = \lim_{\Delta t \to 0} \begin{cases} 1/\Delta t & \text{if } -\Delta t/2 < t < \Delta t/2 \\ 0 & \text{otherwise} \end{cases} \tag{A.28}$$

or a Gaussian function

$$\delta(t) = \lim_{\Delta t \to 0} \frac{1}{\sqrt{2\pi}\,\Delta t} \exp\left[-\frac{1}{2} \left(\frac{t}{\Delta t} \right)^2 \right] . \tag{A.29}$$

δ function definition

It is most often expressed as

$$\delta(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} d\omega \, \exp(i\omega t) . \tag{A.30}$$

This underlies the inverse Fourier transform, as discussed below.

Eigenfunctions

The functional analog of the eigenvector (equation A.12) is the eigenfunction $e(t)$ that satisfies

$$\int dt' \, W(t, t') e(t') = \lambda e(t) . \tag{A.31}$$

For translationally invariant integral operators, $W(t, t') = K(t - t')$, the eigenfunctions are complex exponentials,

$$\int dt' \, K(t - t') \exp(i\omega t') = \left(\int d\tau \, K(\tau) \exp(-i\omega\tau) \right) \exp(i\omega t) , \tag{A.32}$$

as can be seen by making the change of variables $\tau = t - t'$. Here $i = \sqrt{-1}$, and the complex exponential is defined by the identity

complex exponential

$$\exp(i\omega t) = \cos(\omega t) + i \sin(\omega t) . \tag{A.33}$$

Comparing equations A.31 and A.32, we see that the eigenvalue for this eigenfunction is

$$\lambda(\omega) = \int d\tau \, K(\tau) \exp(-i\omega\tau) . \tag{A.34}$$

In this case, the continuous label ω takes the place of the discrete label μ used to identify the different eigenvalues of a matrix.

A functional analog of expanding a vector using eigenvectors as a basis (equation A.14) is the inverse Fourier transform, which expresses a function in an expansion using complex exponential eigenfunctions as a basis. The analog of equation A.16 for determining the coefficient functions of this expansion is the Fourier transform.

The kth derivative of g with respect to α, evaluated at the point $\alpha = 0$, is

$$\left.\frac{d^k g}{d\alpha^k}\right|_{\alpha=0} = \sum_{n=0}^{\infty} \frac{n^k (\mathrm{r}T)^n}{n!} \exp(-\mathrm{r}T) , \tag{1.48}$$

so once we have computed g, we need to calculate only its first and second derivatives to determine the sums we need. Rearranging the terms a bit, and recalling that $\exp(z) = \sum z^n/n!$, we find

$$g(\alpha) = \exp(-\mathrm{r}T) \sum_{n=0}^{\infty} \frac{\left(\mathrm{r}T \exp(\alpha)\right)^n}{n!} = \exp(-\mathrm{r}T) \exp\left(\mathrm{r}Te^{\alpha}\right) . \tag{1.49}$$

The derivatives are then

$$\frac{dg}{d\alpha} = \mathrm{r}Te^{\alpha} \exp(-\mathrm{r}T) \exp(\mathrm{r}Te^{\alpha}) \tag{1.50}$$

and

$$\frac{d^2 g}{d\alpha^2} = (\mathrm{r}Te^{\alpha})^2 \exp(-\mathrm{r}T) \exp(\mathrm{r}Te^{\alpha}) + \mathrm{r}Te^{\alpha} \exp(-\mathrm{r}T) \exp(\mathrm{r}Te^{\alpha}) . \tag{1.51}$$

Evaluating these at $\alpha = 0$ and putting the results into equations 1.45 and 1.46 gives the results $\langle n \rangle = \mathrm{r}T$ and $\sigma_n^2(T) = (\mathrm{r}T)^2 + \mathrm{r}T - (\mathrm{r}T)^2 = \mathrm{r}T$.

C: Inhomogeneous Poisson Statistics

The probability density for a particular spike sequence with spike times t_i for $i = 1, 2, \ldots, n$ is obtained from the corresponding probability distribution by multiplying the probability that the spikes occur when they do by the probability that no other spikes occur. We begin by computing the probability that no spikes are generated during the time interval from t_i to t_{i+1} between two adjacent spikes. We determine this by dividing the interval into M bins of size Δt and setting $M\Delta t = t_{i+1} - t_i$. We will ultimately take the limit $\Delta t \to 0$. The firing rate during bin m within this interval is $\mathrm{r}(t_i + m\Delta t)$. Because the probability of firing a spike in this bin is $\mathrm{r}(t_i + m\Delta t)\Delta t$, the probability of not firing a spike is $1 - \mathrm{r}(t_i + m\Delta t)\Delta t$. To have no spikes during the entire interval, we must string together M such bins, and the probability of this occurring is the product of the individual probabilities,

$$P[\text{no spikes}] = \prod_{m=1}^{M} \left(1 - \mathrm{r}(t_i + m\Delta t)\Delta t\right) . \tag{1.52}$$

We evaluate this expression by taking its logarithm,

$$\ln P[\text{no spikes}] = \sum_{m=1}^{M} \ln\left(1 - \mathrm{r}(t_i + m\Delta t)\Delta t\right) , \tag{1.53}$$

sider this case. The eigenvalues of a symmetric matrix are real, and the eigenvectors are real and orthogonal (or can be made orthogonal in the case of degeneracy). This means that, if they are normalized to unit length, the eigenvectors satisfy

orthonormal eigenvectors

$$\mathbf{e}_\mu \cdot \mathbf{e}_\nu = \delta_{\mu\nu} . \tag{A.15}$$

To derive this result we note that, if $\mathbf{W}$ is a symmetric matrix, we can write $\mathbf{e}_\mu \cdot \mathbf{W} = \mathbf{W} \cdot \mathbf{e}_\mu = \lambda_\mu \mathbf{e}_\mu$. Therefore, allowing the matrix to act in both directions, we find $\mathbf{e}_\nu \cdot \mathbf{W} \cdot \mathbf{e}_\mu = \lambda_\mu \mathbf{e}_\nu \cdot \mathbf{e}_\mu = \lambda_\nu \mathbf{e}_\nu \cdot \mathbf{e}_\mu$. If $\lambda_\mu \neq \lambda_\nu$, this requires $\mathbf{e}_\nu \cdot \mathbf{e}_\mu = 0$. For orthogonal and normalized (orthonormal) eigenvectors, the coefficients in equation A.14 take the values

$$c_\mu = \mathbf{v} \cdot \mathbf{e}_\mu . \tag{A.16}$$

Let $\mathbf{E} = (\mathbf{e}_1\ \mathbf{e}_2\ \ldots\ \mathbf{e}_N)$ be an N by N matrix with columns formed from the orthonormal eigenvectors of a symmetric matrix. From equation A.15, this satisfies $[\mathbf{E}^{\mathrm{T}} \cdot \mathbf{E}]_{\mu\nu} = \mathbf{e}_\mu \cdot \mathbf{e}_\nu = \delta_{\mu\nu}$. Thus, $\mathbf{E}^{\mathrm{T}} = \mathbf{E}^{-1}$, making $\mathbf{E}$ an orthogonal matrix. $\mathbf{E}$ generates a transformation from the original matrix $\mathbf{W}$ to a diagonal form, which is called matrix diagonalization,

matrix diagonalization

$$\mathbf{E}^{-1} \cdot \mathbf{W} \cdot \mathbf{E} = \mathrm{diag}(\lambda_1, \ldots, \lambda_N) . \tag{A.17}$$

Conversely,

$$\mathbf{W} = \mathbf{E} \cdot \mathrm{diag}(\lambda_1, \ldots, \lambda_N) \cdot \mathbf{E}^{-1} . \tag{A.18}$$

The transformation to and back from a diagonal form is extremely useful because computations with diagonal matrices are easy. Defining $\mathbf{L} = \mathrm{diag}(\lambda_1, \ldots, \lambda_N)$, we find, for example, that

$$\begin{aligned} \mathbf{W}^n &= (\mathbf{E} \cdot \mathbf{L} \cdot \mathbf{E}^{-1}) \cdot (\mathbf{E} \cdot \mathbf{L} \cdot \mathbf{E}^{-1}) \cdots (\mathbf{E} \cdot \mathbf{L} \cdot \mathbf{E}^{-1}) \\ &= \mathbf{E} \cdot \mathbf{L}^n \cdot \mathbf{E}^{-1} = \mathbf{E} \cdot \mathrm{diag}(\lambda_1^n, \ldots, \lambda_N^n) \cdot \mathbf{E}^{-1} . \end{aligned} \tag{A.19}$$

This result serves as a basis for defining functions of matrices. For any function f that can be written as a power or expanded in a power series (including, for example, exponentials and logarithms),

$$f(\mathbf{W}) = \mathbf{E} \cdot \mathrm{diag}(f(\lambda_1), \ldots, f(\lambda_N)) \cdot \mathbf{E}^{-1} . \tag{A.20}$$

Functional Analogs

A function $v(t)$ can be treated as if it were a vector with a continuous label. In other words, the function value $v(t)$ parameterized by the continuously varying argument t takes the place of the component v_a labeled by the integer-valued index a. In applying this analogy, sums over a for vectors are replaced by integrals over t for functions, $\sum_a \to \int dt$. For example, the functional analog of the squared norm and dot product are

functions as vectors

$$\int_{-\infty}^{\infty} dt\, v^2(t) \quad \text{and} \quad \int_{-\infty}^{\infty} dt\, v(t) u(t) . \tag{A.21}$$

describes the analysis of spikes and the relationships between neural responses and stimuli, and is a general reference for material we present in chapters 1–4. **Gabbiani & Koch (1998)** provides another account of some of this material. The mathematics underlying point processes, the natural statistical model for spike sequences, is found in **Cox (1962)** and **Cox & Isham (1980)**, including the relationship between the Fano factor and the coefficient of variation. A general analysis of histogram representations appears in **Scott (1992)**, and white-noise and filtering techniques (our analysis of which continues in chapter 2) are described in de Boer & Kuyper (1968), **Marmarelis & Marmarelis (1978)**, and **Wiener (1958)**. Berry & Meister (1998) discuss the effects of refractoriness on patterns of spiking.

In chapters 1 and 3, we discuss two systems associated with studies of spike encoding; the H1 neuron in the visual system of flies, reviewed by **Rieke et al. (1997)**, and area MT of monkeys, discussed by Parker & Newsome (1998). **Wandell (1995)** introduces orientation and disparity tuning, relevant to examples presented in this chapter.

which has components $W_{ab} = h_a \delta_{ab}$ for some set of h_a, $a = 1, 2, \ldots, N$.

The transpose of an N_r by N_c matrix $\mathbf{W}$ is an N_c by N_r matrix $\mathbf{W}^T$ with elements $[\mathbf{W}^T]_{ab} = W_{ba}$. The transpose of a column vector is a row vector, $\mathbf{v}^T = (v_1 v_2 \ldots v_N)$. This is distinguished by the absence of commas from $(v_1, v_2, \ldots, v_N)$ which, for us, is a listing of the components of a column vector. In the following table, we define a number of operations involving vectors and matrices. In the definitions, we provide our notation and the corresponding expressions in terms of vector components and matrix elements. We also provide the conventional matrix notation for these quantities as well as the notation used by MATLAB®, a computer software package commonly used to perform these operations numerically. For the MATLAB® notation (which does not use bold or italic symbols), we denote two column vectors by u and v, assuming they are defined within MATLAB® by instructions such as v =[v(1) v(2) ... v(N)]'. *transpose*

Quantity	Definition	Matrix	MATLAB®
norm	$\lvert\mathbf{v}\rvert^2 = \mathbf{v}\cdot\mathbf{v} = \sum_a v_a^2$	$\mathbf{v}^T\mathbf{v}$	v'*v
dot product	$\mathbf{v}\cdot\mathbf{u} = \sum_a v_a u_a$	$\mathbf{v}^T\mathbf{u}$	v'*u
outer product	$[\mathbf{vu}]_{ab} = v_a u_b$	$\mathbf{v}\mathbf{u}^T$	v*u'
matrix-vector product	$[\mathbf{W}\cdot\mathbf{v}]_a = \sum_b W_{ab} v_b$	$\mathbf{Wv}$	W*v
vector-matrix product	$[\mathbf{v}\cdot\mathbf{W}]_a = \sum_b v_b W_{ba}$	$\mathbf{v}^T\mathbf{W}$	v'*W
quadratic form	$\mathbf{v}\cdot\mathbf{W}\cdot\mathbf{u} = \sum_{ab} v_a W_{ab} u_b$	$\mathbf{v}^T\mathbf{Wu}$	v'*W*u
matrix-matrix product	$[\mathbf{W}\cdot\mathbf{M}]_{ab} = \sum_c W_{ac} M_{cb}$	$\mathbf{WM}$	W*M
transpose	$[\mathbf{W}^T]_{ab} = W_{ba}$	$\mathbf{W}^T$	W'

Several important definitions for square matrices are given below.

Operation	Notation	Definition	MATLAB®
inverse	$\mathbf{W}^{-1}$	$\mathbf{W}\cdot\mathbf{W}^{-1} = \mathbf{I}$	`inv`(W)
trace	tr$\mathbf{W}$	$\sum_a W_{aa}$	`trace`(W)
determinant	det$\mathbf{W}$	see references	`det`(W)

A square matrix has an inverse only if its determinant is nonzero. Square matrices with certain properties are given special names (table below).

Property	Definition
symmetric	$\mathbf{W}^T = \mathbf{W}$ or $W_{ba} = W_{ab}$
orthogonal	$\mathbf{W}^T = \mathbf{W}^{-1}$ or $\mathbf{W}^T\cdot\mathbf{W} = \mathbf{I}$
positive-definite	$\mathbf{v}\cdot\mathbf{W}\cdot\mathbf{v} > 0$ for all $\mathbf{v} \neq \mathbf{0}$
Töplitz	$W_{ab} = f(a-b)$

curves. The activity of a neuron at time t typically depends on the behavior of the stimulus over a period of time starting a few hundred milliseconds prior to t and ending perhaps tens of milliseconds before t. Reverse-correlation methods can be used to construct a more accurate model that includes the effects of the stimulus over such an extended period of time. The basic problem is to construct an estimate $r_{est}(t)$ of the firing rate $r(t)$ evoked by a stimulus $s(t)$. The simplest way to construct an estimate is to assume that the firing rate at any given time can be expressed as a weighted sum of the values taken by the stimulus at earlier times. Because time is a continuous variable, this "sum" actually takes the form of an integral, and we write

firing rate estimate $r_{est}(t)$

$$r_{est}(t) = r_0 + \int_0^\infty d\tau\, D(\tau) s(t-\tau)\,. \tag{2.1}$$

The term r_0 accounts for any background firing that may occur when $s = 0$. $D(\tau)$ is a weighting factor that determines how strongly, and with what sign, the value of the stimulus at time $t - \tau$ affects the firing rate at time t. Note that the integral in equation 2.1 is a linear filter of the same form as the expressions used to compute $r_{approx}(t)$ in chapter 1.

As discussed in chapter 1, sensory systems tend to adapt to the absolute intensity of a stimulus. It is easier to account for the responses to fluctuations of a stimulus around some mean background level than it is to account for adaptation processes. We therefore assume throughout this chapter that the stimulus parameter $s(t)$ has been defined with its mean value subtracted out. This means that the time integral of $s(t)$ over the duration of a trial is 0.

We have provided a heuristic justification for the terms in equation 2.1 but, more formally, they correspond to the first two terms in a systematic expansion of the response in powers of the stimulus. Such an expansion, called a Volterra expansion, is the functional equivalent of the Taylor series expansion used to generate power series approximations of functions. For the case we are considering, it takes the form

Volterra expansion

$$r_{est}(t) = r_0 + \int d\tau\, D(\tau) s(t-\tau) + \int d\tau_1 d\tau_2\, D_2(\tau_1, \tau_2) s(t-\tau_1) s(t-\tau_2) + \int d\tau_1 d\tau_2 d\tau_3\, D_3(\tau_1, \tau_2, \tau_3) s(t-\tau_1) s(t-\tau_2) s(t-\tau_3) + \ldots . \tag{2.2}$$

This series was rearranged by Wiener to make the terms easier to compute. The first two terms of the Volterra and Wiener expansions take the same mathematical forms and are given by the two expressions on the right side of equation 2.1. For this reason, D is called the first Wiener kernel, the linear kernel, or, when higher-order terms (terms involving more than one factor of the stimulus) are not being considered, simply the kernel.

Wiener expansion

Wiener kernel

To construct an estimate of the firing rate based on equation 2.1, we choose the kernel D to minimize the squared difference between the estimated response to a stimulus and the actual measured response averaged over the

Mathematical Appendix

This book assumes a familiarity with basic methods of linear algebra, differential equations, and probability theory, as covered in standard texts. Here, we describe the notation we use and briefly sketch highlights of various techniques. The references in the bibliography at the end of this appendix provides further information.

A.1 Linear Algebra

An operation $\mathcal{O}$ on a quantity z is called linear if, applied to any two instances z_1 and z_2, $\mathcal{O}(\alpha z_1 + \beta z_2) = \alpha \mathcal{O}(z_1) + \beta \mathcal{O}(z_2)$ for any constants α and β. In this section, we consider linear operations on vectors and functions. We define a vector $\mathbf{v}$ as an array of N numbers $(v_1, v_2, \ldots, v_N)$. The numbers v_a for $a = 1, 2, \ldots, N$ are called the components of the vector. These are sometimes listed in a single N-row column

linear operator

vector $\mathbf{v}$

$$\mathbf{v} = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{pmatrix} . \tag{A.1}$$

When necessary, we write component a of $\mathbf{v}$ as $[\mathbf{v}]_a = v_a$. We use $\mathbf{0}$ to denote the vector with all its components equal to 0. Spatial vectors, which are related to displacements in space, are a special case, and we denote them by $\vec{v}$ with components v_x and v_y in two-dimensional space or v_x, v_y, and v_z in three-dimensional space.

zero vector $\mathbf{0}$

spatial vector $\vec{v}$

The length or norm of $\mathbf{v}$, $|\mathbf{v}|$, when squared, can be written as a dot product,

norm

$$|\mathbf{v}|^2 = \mathbf{v} \cdot \mathbf{v} = \sum_{a=1}^{N} v_a^2 = v_1^2 + v_2^2 + \ldots + v_N^2 . \tag{A.2}$$

The dot product of two different N-component vectors, $\mathbf{v}$ and $\mathbf{u}$, is

dot product

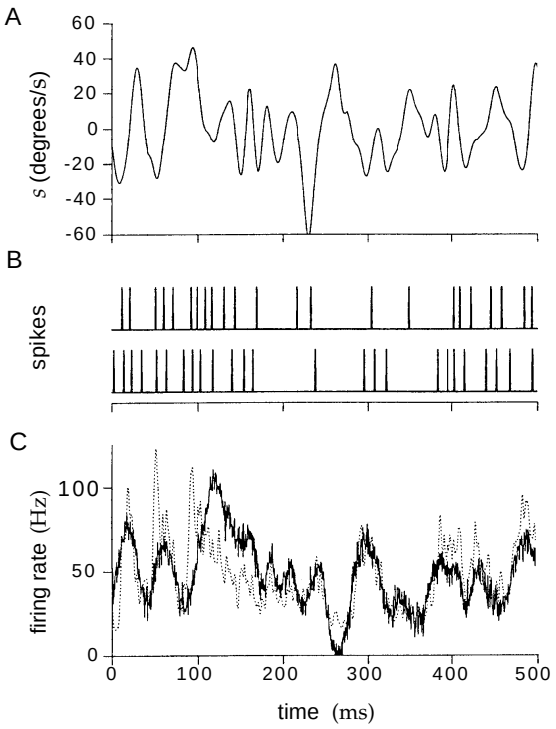

Figure 2.1 Prediction of the firing rate for an H1 neuron responding to a moving visual image. (A) The velocity of the image used to stimulate the neuron. (B) Two of the 100 spike sequences used in this experiment. (C) Comparison of the measured and computed firing rates. The dashed line is the firing rate extracted directly from the spike trains. The solid line is an estimate of the firing rate constructed by linearly filtering the stimulus with an optimal kernel. (Adapted from Rieke et al., 1997.)

The Most Effective Stimulus

Neuronal selectivity is often characterized by describing stimuli that evoke maximal responses. The reverse-correlation approach provides a basis for this procedure by relating the optimal kernel for firing-rate estimation to the stimulus predicted to evoke the maximum firing rate, subject to a constraint. A constraint is essential because the linear estimate in equation 2.1 is unbounded. The constraint we use is that the time integral of the square of the stimulus over the duration of the trial is held fixed. We call this integral the stimulus energy. The stimulus for which equation 2.1 predicts the maximum response at some fixed time subject to this constraint, is computed in appendix B. The result is that the stimulus producing the maximum response is proportional to the optimal linear kernel or, equivalently, to the white-noise spike-triggered average stimulus. This is an important result because in cases where a white-noise analysis has not been done, we may still have some idea what stimulus produces the maximum response.

The maximum stimulus analysis provides an intuitive interpretation of

interpretations of ICA. Multi-resolution decompositions were introduced into computer vision in Witkin (1983) and Burt & Adelson (1983). Wavelet analysis is reviewed in Daubechies (1992), **Simoncelli et al. (1992)**, and **Mallat (1998)**.

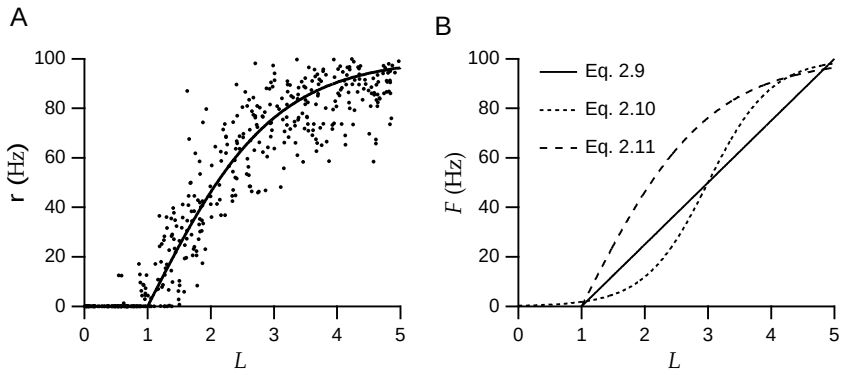

Figure 2.2 (A) A graphical procedure for determining static nonlinearities. Linear estimates L and actual firing rates r are plotted (solid points) and fitted by the function $F(L)$ (solid line). (B) Different static nonlinearities used in estimating neural responses. L is dimensionless, and equations 2.9, 2.10, and 2.11 have been used with $G = 25$ Hz, $L_0 = 1$, $L_{1/2} = 3$, $r_{\max} = 100$ Hz, $g_1 = 2$, and $g_2 = 1/2$.

writing

$$F(L) = G[L - L_0]_+ , \tag{2.9}$$

threshold function

where L_0 is the threshold value that L must attain before firing begins. Above the threshold, the firing rate is a linear function of L, with G acting as the constant of proportionality. Half-wave rectification is a special case of this with $L_0 = 0$. That this function does not saturate is not a problem if large stimulus values are avoided. If needed, a saturating nonlinearity can be included in F, and a sigmoidal function is often used for this purpose,

rectification

sigmoidal function

$$F(L) = \frac{\mathsf{r}_{\max}}{1 + \exp\left(g_1(L_{1/2} - L)\right)} . \tag{2.10}$$

Here $r_{\max}$ is the maximum possible firing rate, $L_{1/2}$ is the value of L for which F achieves half of this maximal value, and g_1 determines how rapidly the firing rate increases as a function of L. Another choice that combines a hard threshold with saturation uses a rectified hyperbolic tangent function,

$$F(L) = \mathsf{r}_{\max}[\tanh\left(g_2(L - L_0)\right)]_+ , \tag{2.11}$$

where $r_{\max}$ and g_2 play similar roles as in equation 2.10, and L_0 is the threshold. Figure 2.2B shows the different nonlinear functions that we have discussed.

Although the static nonlinearity can be any function, the estimate of equation 2.8 is still restrictive because it allows for no dependence on weighted autocorrelations of the stimulus or other higher-order terms in the Volterra series. Furthermore, once the static nonlinearity is introduced, the linear kernel derived from equation 2.4 is no longer optimal because it was chosen to minimize the squared error of the linear estimate $\mathsf{r}_{\text{est}}(t) = \mathsf{r}_0 + L(t)$,

10.6 Appendix

Summary of Causal Models

Model	Generative Model	Recognition Model	Learning Rules
mixture of Gaussians	$P[v;\mathcal{G}] = \gamma_v$ $P[\mathbf{u}\|v;\mathcal{G}] = \mathcal{N}(\mathbf{u};\mathbf{g}_v,\Sigma_v)$	$P[v\|\mathbf{u};\mathcal{G}] \propto \gamma_v \mathcal{N}(\mathbf{u};\mathbf{g}_v,\Sigma_v)$	$\mu_v = \langle P[v\|\mathbf{u};\mathcal{G}]\rangle$ $\mathbf{g}_v = \langle P[v\|\mathbf{u};\mathcal{G}]\mathbf{u}\rangle/\gamma_v$ $\Sigma_v = \langle P[v\|\mathbf{u};\mathcal{G}]\|\mathbf{u}-\mathbf{g}_v\|^2\rangle/(N_u\gamma_v)$
factor analysis	$P[\mathbf{v};\mathcal{G}] = \mathcal{N}(\mathbf{v};\mathbf{0},1)$ $P[\mathbf{u}\|\mathbf{v};\mathcal{G}] = \mathcal{N}(\mathbf{u};\mathbf{G}\cdot\mathbf{v},\mathbf{\Sigma})$ $\mathbf{\Sigma} = \mathrm{diag}\left(\Sigma_1,\Sigma_2,\ldots,\Sigma_{N_u}\right)$	$P[\mathbf{v}\|\mathbf{u};\mathcal{G}] = \mathcal{N}(\mathbf{v};\mathbf{W}\cdot\mathbf{u},\mathbf{\Psi})$ $\mathbf{\Psi} = \left(\mathbf{I}+\mathbf{G}^{\mathrm{T}}\cdot\mathbf{\Sigma}^{-1}\cdot\mathbf{G}\right)^{-1}$ $\mathbf{W} = \mathbf{\Psi}\cdot\mathbf{G}^{\mathrm{T}}\cdot\mathbf{\Sigma}^{-1}$	$\mathbf{G} = \mathbf{C}\cdot\mathbf{W}^{\mathrm{T}}\cdot(\mathbf{W}\cdot\mathbf{C}\cdot\mathbf{W}^{\mathrm{T}}+\mathbf{\Psi})^{-1}$ $\mathbf{\Sigma} = \mathrm{diag}\left(\mathbf{G}\cdot\mathbf{\Psi}\cdot\mathbf{G}^{\mathrm{T}} + (\mathbf{I}-\mathbf{G}\cdot\mathbf{W})\cdot\mathbf{C}\cdot(\mathbf{I}-\mathbf{G}\cdot\mathbf{W})^{\mathrm{T}}\right)$ $\mathbf{C} = \langle\mathbf{u}\mathbf{u}\rangle$
principal components analysis	$P[\mathbf{v};\mathcal{G}] = \mathcal{N}(\mathbf{v};\mathbf{0},1)$ $\mathbf{u} = \mathbf{G}\cdot\mathbf{v}$	$\mathbf{v} = \mathbf{W}\cdot\mathbf{u}$ $\mathbf{W} = (\mathbf{G}^{\mathrm{T}}\cdot\mathbf{G})^{-1}\cdot\mathbf{G}^{\mathrm{T}}$	$\mathbf{G} = \mathbf{C}\cdot\mathbf{W}^{\mathrm{T}}\cdot(\mathbf{W}\cdot\mathbf{C}\cdot\mathbf{W}^{\mathrm{T}})^{-1}$ $\mathbf{C} = \langle\mathbf{u}\mathbf{u}\rangle$
sparse coding	$P[\mathbf{v};\mathcal{G}] \propto \prod_a \exp(g(v_a))$ $P[\mathbf{u}\|\mathbf{v};\mathcal{G}] = \mathcal{N}(\mathbf{u};\mathbf{G}\cdot\mathbf{v},\Sigma)$	$\mathbf{G}^{\mathrm{T}}\cdot(\mathbf{u}-\mathbf{G}\cdot\mathbf{v}) + \mathbf{g}'(\mathbf{v}) = 0$	$\mathbf{G} \to \mathbf{G} + \epsilon(\mathbf{u}-\mathbf{G}\cdot\mathbf{v})\mathbf{v}$ $\left(\sum_b G_{ba}^2\right) \to \left(\sum_b G_{ba}^2\right)\left(\langle v_a^2\rangle - \langle v_a\rangle^2/\sigma^2\right)^{0.01}$
independent components analysis	$P[\mathbf{v};\mathcal{G}] \propto \prod_a \exp(g(v_a))$ $\mathbf{u} = \mathbf{G}\cdot\mathbf{v}$	$\mathbf{v} = \mathbf{W}\cdot\mathbf{u}$ $\mathbf{W} = \mathbf{G}^{-1}$	$\mathbf{W}_{ab} \to \mathbf{W}_{ab} + \epsilon\left(\mathbf{W}_{ab} + g'(v_a)[\mathbf{v}\cdot\mathbf{W}]_b\right)$ $g'(v) = -\tanh(v)$ if $g(v) = -\ln\cosh(v)$
binary Helmholtz machine	$P[\mathbf{v};\mathcal{G}] = \prod_a \left(f(g_a)\right)^{v_a}\left(1-f(g_a)\right)^{1-v_a}$ $P[\mathbf{u}\|\mathbf{v};\mathcal{G}] = \prod_b \left(f_b(\mathbf{h}+\mathbf{G}\cdot\mathbf{v})\right)^{u_b} \times \left(1-f_b(\mathbf{h}+\mathbf{G}\cdot\mathbf{v})\right)^{1-u_b}$ $f_b(\mathbf{h}+\mathbf{G}\cdot\mathbf{v}) = f\left(h_b + [\mathbf{G}\cdot\mathbf{v}]_b\right)$	$Q[\mathbf{v};\mathbf{u},\mathcal{W}] = \prod_a \left(f_a(\mathbf{w}+\mathbf{W}\cdot\mathbf{u})\right)^{v_a} \times \left(1-f_a(\mathbf{w}+\mathbf{W}\cdot\mathbf{u})\right)^{1-v_a}$ $f_a(\mathbf{w}+\mathbf{W}\cdot\mathbf{u}) = f\left(w_a + [\mathbf{W}\cdot\mathbf{u}]_a\right)$	wake: $\mathbf{u} \sim P[\mathbf{u}]$, $\mathbf{v} \sim Q[\mathbf{v};\mathbf{u},\mathcal{W}]$ $\mathbf{g} \to \mathbf{g} + \epsilon(\mathbf{v}-\mathbf{f}(\mathbf{g}))$ $\mathbf{h} \to \mathbf{h} + \epsilon(\mathbf{u}-\mathbf{f}(\mathbf{h}+\mathbf{G}\cdot\mathbf{v}))$ $\mathbf{G} \to \mathbf{G} + \epsilon(\mathbf{u}-\mathbf{f}(\mathbf{h}+\mathbf{G}\cdot\mathbf{v}))\mathbf{v}$ sleep: $\mathbf{v} \sim P[\mathbf{v};\mathcal{G}]$, $\mathbf{u} \sim P[\mathbf{u}\|\mathbf{v};\mathcal{G}]$ $\mathbf{w} \to \mathbf{w} + \epsilon(\mathbf{v}-\mathbf{f}(\mathbf{w}+\mathbf{W}\cdot\mathbf{u}))$ $\mathbf{W} \to \mathbf{W} + \epsilon(\mathbf{v}-\mathbf{f}(\mathbf{w}+\mathbf{W}\cdot\mathbf{u}))\mathbf{u}$

Table 1: All models are discussed in detail in the text, and the forms quoted are just for the simplest cases. $\mathcal{N}(\mathbf{u};\mathbf{g},\mathbf{\Sigma})$ is a multivariate Gaussian distribution with mean $\mathbf{g}$ and covariance matrix $\mathbf{\Sigma}$ (for $\mathcal{N}(\mathbf{u};\mathbf{g},\Sigma)$, the variance of each component is Σ). For the sparse coding network, σ^2 is a target for the variances of each output unit. For the Helmholtz machine, $f(c) = 1/(1+\exp(-c))$, and the symbol $\sim$ indicates that the indicated variable is drawn from the indicated distribution. Other symbols and distributions are defined in the text.

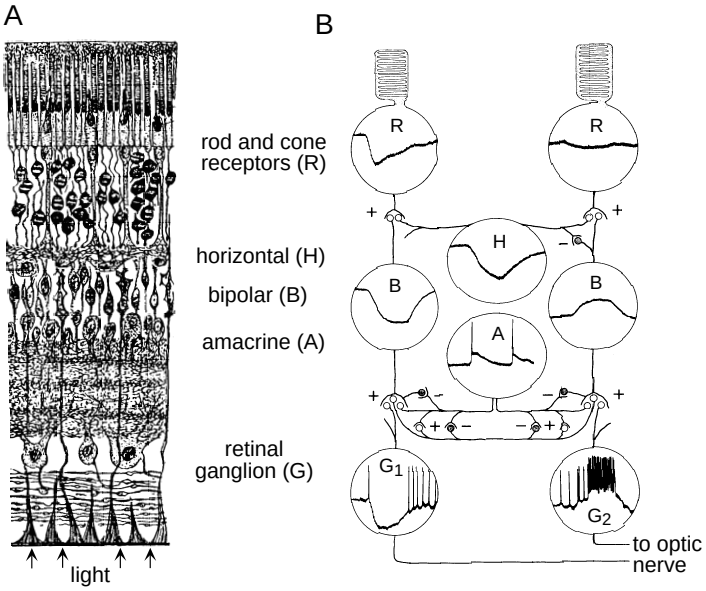

Figure 2.4 (A) An anatomical diagram of the circuitry of the retina of a dog. Cell types are identified at right. In the intact eye, counterintuitively, light enters through the side opposite from the photoreceptors. (B) Intracellular recordings from retinal neurons of the mud puppy responding to a flash of light lasting for 1 s. In the column of cells on the left side of the diagram, the resulting hyperpolarizations are about 4 mV in the rod and retinal ganglion cells, and 8 mV in the bipolar cell. Pluses and minuses represent excitatory and inhibitory synapses, respectively. (A adapted from Nicholls et al., 1992; drawing from Cajal, 1911. B data from Werblin and Dowling 1969; figure adapted from Dowling, 1992.)

early stages of the visual system. The conversion of a light stimulus into an electrical signal and ultimately an action potential sequence occurs in the retina. Figure 2.4A is an anatomical diagram showing the five principal cell types of the retina, and figure 2.4B is a rough circuit diagram. In the retina, light is first converted into an electrical signal by a phototransduction cascade within rod and cone photoreceptor cells. Figure 2.4B shows intracellular recordings made in neurons of the retina of a mud puppy (an amphibian). The stimulus used for these recordings was a flash of light falling primarily in the region of the photoreceptor at the left of figure 2.4B. The rod cells, especially the one on the left side of figure 2.4B, are hyperpolarized by the light flash. This electrical signal is passed along to bipolar and horizontal cells through synaptic connections. Note that in one of the bipolar cells, the signal has been inverted, leading to depolarization. These smoothly changing membrane potentials provide a graded representation of the light intensity during the flash. This form of coding is adequate for signaling within the retina, where distances are small. However, it is inadequate for the task of conveying information from the retina to the brain.

retinal ganglion cells

The output neurons of the retina are the retinal ganglion cells, whose axons form the optic nerve. As seen in figure 2.4B, the subthreshold potentials

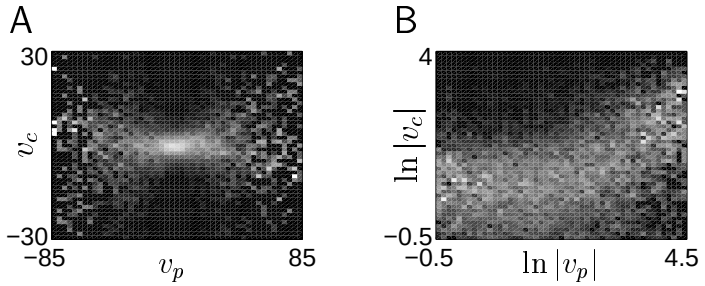

Figure 10.12 (A) Gray-scale plot of the conditional distribution of the output of a filter at the finest spatial scale (v_c) given the output of a coarser filter (v_p) with the same position and orientation (using the picture in figure 10.11A as input data). Each column is separately normalized. The plot has a characteristic bow-tie shape. (B) The same data plotted as the conditional distribution of $\ln|v_c|$ given $\ln|v_p|$. (Adapted from Simoncelli & Adelson, 1990; Simoncelli & Schwartz, 1999.)

the redundant filter outputs wasteful. Figure 10.12 illustrates such an interdependence by showing the conditional distribution for the output v_c of a horizontally tuned filter at a fine scale, given the output v_p of a horizontally tuned unit at the next coarser scale. The plots show gray-scale values of the conditional probability density $p[v_c|v_p]$. The mean of this distribution is roughly 0, but there is a clear correlation between the magnitude of $|v_p|$ and the variance of v_c. This means that structure in the image is coordinated across different spatial scales, so that high outputs from a coarse scale filter are typically accompanied by substantial output (of one sign or the other) at a finer scale. Following Simoncelli (1997), we plot the conditional distribution of $\ln|v_c|$ given $\ln|v_p|$ in figure 10.12B. For small values of $\ln|v_p|$, the distribution of $\ln|v_c|$ is flat, but for larger values of $\ln|v_p|$ the growth in the value of $|v_c|$ is clear.

The interdependence shown in figure 10.12 suggests a failing of sparse coding to which we have alluded before. Although the prior distribution for sparse coding stipulates independent causes, the causes identified as underlying real images are not independent. The dependence apparent in figure 10.12 can be removed by a nonlinear transformation in which the outputs of the units normalize each other (similar to the model introduced to explain contrast saturation in chapter 2). This transformation can lead to more compact codes for images. However, the general problem suggests that something is amiss with the heuristic of seeking independent causes for representations early in the visual pathway.

The most important dependencies as far as causal models are concerned are those induced by the presence in images of objects with large-scale coordinated structure. Finding and building models of these dependencies is the goal for more sophisticated, hierarchical representational learning schemes aimed ultimately at object recognition within complex visual scenes.

simple and complex cells

tions from different locations within the visual field sum linearly. X cells in the cat retina and LGN, P cells in the monkey retina and LGN, and simple cells in primary visual cortex appear to satisfy this assumption. Other neurons, such as Y cells in the cat retina and LGN, M cells in the monkey retina and LGN, and complex cells in primary visual cortex, do not show linear summation across the spatial receptive field, and nonlinearities must be included in descriptions of their responses. We do this for complex cells later in this chapter.

A first step in studying the selectivity of any neuron is to identify the types of stimuli that evoke strong responses. Retinal ganglion cells and LGN neurons have similar selectivities and respond best to circular spots of light surrounded by darkness or dark spots surrounded by light. In primary visual cortex, many neurons respond best to elongated light or dark bars or to boundaries between light and dark regions. Gratings with alternating light and dark bands are effective and frequently used stimuli for these neurons.

Many visually responsive neurons react strongly to sudden transitions in the level of image illumination, a temporal analogue of their responsiveness to light-dark spatial boundaries. Static images are not very effective at evoking visual responses. In awake animals, images are constantly kept in motion across the retina by eye movements. In experiments in which the eyes are fixed, moving light bars and gratings, or gratings undergoing periodic light-dark reversals (called counterphase gratings) are more effective stimuli than static images. Some neurons in primary visual cortex are directionally selective; they respond more strongly to stimuli moving in one direction than in the other.

To streamline the discussion in this chapter, we consider only gray-scale images, although the methods presented can be extended to include color. We also restrict the discussion to two-dimensional visual images, ignoring how visual responses depend on viewing distance and encode depth. In discussing the response properties of retinal, LGN, and V1 neurons, we do not follow the path of the visual signal, nor the historical order of experimentation, but instead begin with primary visual cortex and then move back to the LGN and retina. In this chapter, the emphasis is on properties of individual neurons; we discuss encoding by populations of visually responsive neurons in chapter 10.

The Retinotopic Map

A striking feature of most visual areas in the brain, including primary visual cortex, is that the visual world is mapped onto the cortical surface in a topographic manner. This means that neighboring points in a visual image evoke activity in neighboring regions of visual cortex. The retinotopic map refers to the transformation from the coordinates of the visual world to the corresponding locations on the cortical surface.

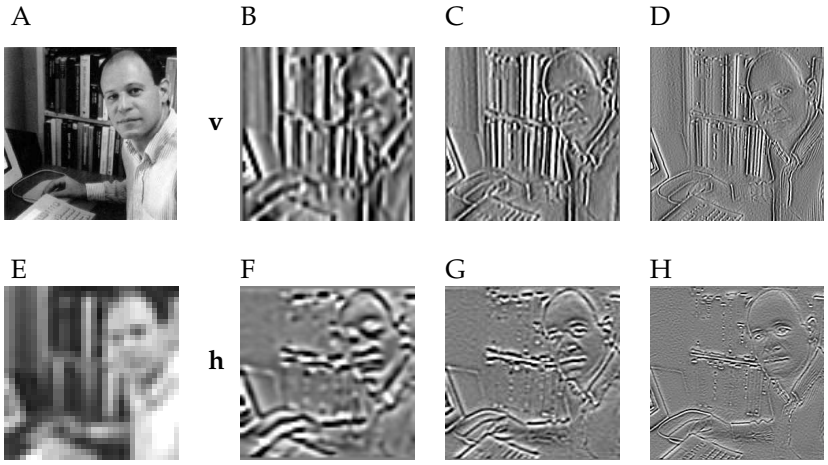

Figure 10.11 Multi-resolution image decomposition. A gray-scale image is decomposed, using the pair of vertical and horizontal filters shown in figure 10.10. (A) The original image. (B-D) The outputs of successively higher spatial frequency, vertically oriented filters translated across the image. (E) The image after passage through a low-pass filter. (F-H) The outputs of successively higher spatial frequency, horizontally oriented filters translated across the image.

provide an inefficient encoding. This is illustrated by the dot-dashed line in figure 10.10B, which shows that the distribution over the values of the input pixels of the image in figure 10.11A is approximately flat or uniform. Up to the usual additive constants related to the precision with which filter outputs are encoded, the contribution to the coding cost from a single unit is the entropy of the probability distribution of its output. The distribution over pixel intensities is flat, which is the maximum entropy distribution for a variable with a fixed range. Encoding the individual pixel values therefore incurs the maximum possible coding cost.

By contrast, the solid line in figure 10.10B shows the distribution of the outputs of the finest scale vertically and horizontally tuned filters (figures 10.11D and 10.11H) in response to figure 10.11A. The filter outputs have a sparse distribution similar to the double exponential distribution in figure 10.5B. This distribution has significantly lower entropy than the uniform distribution, so the filter outputs provide a more efficient encoding than pixel values.

In making these statements about the distributions of activities, we are equating the output distribution of a filter applied at many locations on a single image with the output distribution of a filter applied at a fixed location on many images. This assumes spatial translational invariance of the ensemble of visual images.

Images represented by multi-resolution filters can be further compressed by retaining only approximate values of the filter outputs. This is called lossy coding and may consist of reporting filter outputs as integer multiples of a basic unit. Making the multi-resolution code for an image lossy

lossy coding

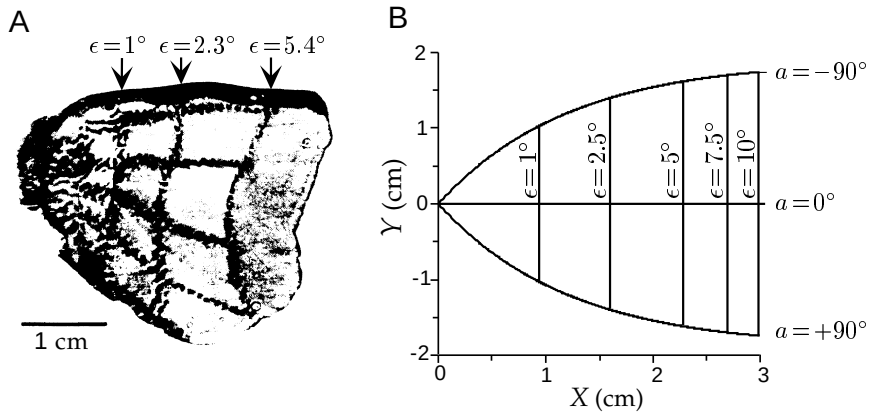

Figure 2.7 (A) An autoradiograph of the posterior region of the primary visual cortex from the left side of a macaque monkey brain. The pattern is a radioactive trace of the activity evoked by an image like that in figure 2.6B. The vertical lines correspond to circles at eccentricities of 1°, 2.3°, and 5.4°, and the horizontal lines (from top to bottom) represent radial lines in the visual image at a values of -90°, -45°, 0°, 45°, and 90°. Only the part of cortex corresponding to the central region of the visual field on one side is shown. (B) The mathematical map from the visual coordinates ϵ and a to the cortical coordinates X and Y described by equations 2.15 and 2.17. (A adapted from Tootell et al., 1982.)

Figure 2.7A shows a dramatic illustration of the retinotopic map in the primary visual cortex of a monkey. The pattern on the cortex seen in figure 2.7A was produced by imaging a radioactive analogue of glucose that was taken up by active neurons while a monkey viewed a visual image consisting of concentric circles and radial lines, similar to the pattern in figure 2.6B. The vertical lines correspond to the circles in the image, and the roughly horizontal lines are due to the activity evoked by the radial lines. The fovea is represented at the leftmost pole of this piece of cortex, and eccentricity increases toward the right. Azimuthal angles are positive in the lower half of the piece of cortex shown, and negative in the upper half.

cortical magnification factor

To describe the map in figure 2.7A mathematically, we write the horizontal and vertical coordinates, X and Y, describing points on the cortical surface as functions of the eccentricity ϵ and azimuth a of the corresponding points in the visual field, $X(\epsilon, a)$, $Y(\epsilon, a)$. This map is characterized by local factors, each called a cortical magnification factor, that indicate the relationship between small displacements ΔX, ΔY across the cortex and the corresponding small image displacements $\Delta\epsilon$, Δa. In general, these factors can be derived from the four elements of the Jacobian matrix, $\partial X/\partial\epsilon$, $\partial X/\partial a$, $\partial Y/\partial\epsilon$, and $\partial Y/\partial a$. However, few experiments characterize all four elements, and it is common to include additional constraints.

Figure 2.7B shows an example of such a map. Here, we assume that $X(\epsilon)$ is only a function of eccentricity, and that $Y(\epsilon, a)$ is proportional to the azimuth angle a. Further, we assume that purely radial ($\Delta a = 0$) and purely

v associated with it by the recognition model. In the E phase, samples of both **v** and **u** are drawn from the generative model distributions $P[\mathbf{v}; \mathcal{G}]$ and $P[\mathbf{u}|\mathbf{v}; \mathcal{G}]$, and the recognition parameters $\mathcal{W}$ are changed according to the discrepancy between the sampled cause **v** and the recognition or bottom-up prediction $\mathbf{f}(\mathbf{w} + \mathbf{W} \cdot \mathbf{u})$ of **v** (see the appendix). The rationale for this is that the **v** that was used by the generative model to create **u** is a good choice for its cause in the recognition model.

wake-sleep algorithm

The two phases of learning are sometimes called wake and sleep because learning in the first phase is driven by real inputs **u** from the environment, while learning in the second phase is driven by values **v** and **u** "fantasized" by the generative model. This terminology is based on slightly different principles from the wake and sleep phases of the Boltzmann machine discussed in chapter 8. The sleep phase is only an approximation of the actual E phase, and general conditions under which learning converges appropriately are not known.

10.4 Discussion

Because of the widespread significance of coding, transmitting, storing, and decoding visual images such as photographs and movies, substantial effort has been devoted to understanding the structure of this class of inputs. As a result, visual images provide an ideal testing ground for representational learning algorithms, allowing us to go beyond evaluating the representations they produce solely in terms of the log likelihood and qualitative similarities with cortical receptive fields.

Most modern image (and auditory) processing techniques are based on multi-resolution decompositions. In such decompositions, images are represented by the activity of a population of units with systematically varying spatial frequency and orientation preferences, centered at various locations on the image. The outputs of the representational units are generated by filters (typically linear) that act as receptive fields and are partially localized in both space and spatial frequency. The filters usually have similar underlying forms, but they are cast at different spatial scales and centered at different locations for the different units. Systematic versions of such representations, in forms such as wavelets, are important signal processing tools, and there is an extensive body of theory about their representational and coding qualities. Representation of sensory information in separated frequency bands at different spatial locations has significant psychophysical consequences as well.

The projective fields of the units in the sparse coding network shown in figure 10.7 suggest that they construct something like a multi-resolution decomposition of inputs, with multiple spatial scales, locations, and orientations. Thus, multi-resolution analysis gives us a way to put into sharper focus the issues arising from models such as sparse coding and independent components analysis. After a brief review of multi-resolution decom-

Visual Stimuli

Earlier in this chapter, we used the function $s(t)$ to characterize a time-dependent stimulus. The description of visual stimuli is more complex. Gray-scale images appearing on a two-dimensional surface, such as a video monitor, can be described by giving the luminance, or light intensity, at each point on the screen. These pixel locations are parameterized by Cartesian coordinates x and y, as in the lower panel of figure 2.6A. However, pixel-by-pixel light intensities are not a useful way of parameterizing a visual image for the purposes of characterizing neuronal responses. This is because visually responsive neurons, like many sensory neurons, adapt to the overall level of screen illumination. To avoid dealing with adaptation effects, we describe the stimulus by a function $s(x, y, t)$ that is proportional to the difference between the luminance at the point (x, y) at time t and the average or background level of luminance. Often $s(x, y, t)$ is also divided by the background luminance level, making it dimensionless. The resulting quantity is called the contrast.

contrast

During recordings, visual neurons are usually stimulated by images that vary over both space and time. A commonly used stimulus, the counterphase sinusoidal grating, is described by

counterphase sinusoidal grating

$$s(x, y, t) = A \cos\left(Kx \cos\Theta + Ky \sin\Theta - \Phi\right) \cos(\omega t) . \qquad (2.18)$$

Figure 2.8 shows a similar grating (a spatial square wave is drawn rather than a sinusoid) and illustrates the significance of the parameters K, Θ, Φ, and ω. K and ω are the spatial and temporal frequencies of the grating (these are angular frequencies), Θ is its orientation, Φ is its spatial phase, and A is its contrast amplitude. This stimulus oscillates in both space and time. At any fixed time, it oscillates in the direction perpendicular to the orientation angle Θ as a function of position, with wavelength $2\pi/K$ (figure 2.8A). At any fixed position, it oscillates in time with period $2\pi/\omega$ (figure 2.8B). For convenience, Θ is measured relative to the y axis rather than the x axis, so that a stimulus with $\Theta = 0$ varies in the x, but not in the y, direction. Φ determines the spatial location of the light and dark stripes of the grating. Changing Φ by an amount $\Delta\Phi$ shifts the grating in the direction perpendicular to its orientation by a fraction $\Delta\Phi/2\pi$ of its wavelength. The contrast amplitude A controls the maximum degree of difference between light and dark areas. Because x and y are measured in degrees, K is expressed in the rather unusual units of radians per degree and $K/2\pi$ is typically reported in units of cycles per degree. Φ has units of radians, ω is in radians/s, and $\omega/2\pi$ is in Hz.

spatial frequency K
frequency ω
orientation Θ
spatial phase Φ
amplitude A

Experiments that consider reverse correlation and spike-triggered averages use various types of random and white-noise stimuli in addition to bars and gratings. A white-noise stimulus, in this case, is one that is uncorrelated in both space and time so that

white-noise image

$$\frac{1}{T}\int_0^T dt\, s(x, y, t)s(x', y', t + \tau) = \sigma_s^2 \delta(\tau)\delta(x - x')\delta(y - y') . \qquad (2.19)$$

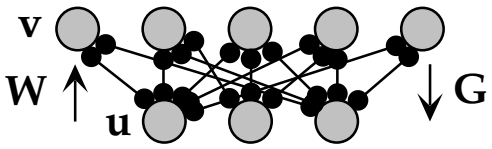

Figure 10.9 Network for the Helmholtz machine. In the bottom-up network, representational units **v** are driven by inputs **u** through feedforward weights **W**. In the top-down network, the inputs are driven by the **v** units through feedback weights **G**.

The Helmholtz Machine

The Helmholtz machine was designed to accommodate hierarchical architectures that construct complex multilayer representations. The model involves two interacting networks, one with parameters $\mathcal{G}$ that is driven in the top-down direction to implement the generative model, and the other, with parameters $\mathcal{W}$, driven bottom-up to implement the recognition model. The parameters are determined by a modified EM algorithm that results in roughly symmetric updates for the two networks.

We consider a simple, two-layer, nonlinear Helmholtz machine (figure 10.9) with binary units, so that u_b and v_a for all b and a take the values 0 or 1. For this model,

$$P[\mathbf{v};\mathcal{G}] = \prod_a \left(f(g_a)\right)^{v_a}\left(1-f(g_a)\right)^{1-v_a} \tag{10.41}$$

$$P[\mathbf{u}|\mathbf{v};\mathcal{G}] = \prod_b \left(f\left(h_b+[\mathbf{G}\cdot\mathbf{v}]_b\right)\right)^{u_b}\left(1-f\left(h_b+[\mathbf{G}\cdot\mathbf{v}]_b\right)\right)^{1-u_b}, \tag{10.42}$$

where g_a is a generative bias weight for output a that controls how frequently $v_a = 1$, h_b is the generative bias weight for u_b, and $f(g) = 1/(1+\exp(-g))$ is the standard sigmoid function. The generative model is thus parameterized by $\mathcal{G} = (\mathbf{g}, \mathbf{h}, \mathbf{G})$. According to these distributions, the components of **v** are mutually independent, and the components of **u** are independent given a fixed value of **v**.

The generative model is noninvertible in this case, so an approximate recognition distribution must be constructed. This uses a form similar to equation 10.42, except with bottom-up weights **W** and biases **w**,

$$Q[\mathbf{v};\mathbf{u},\mathcal{W}] = \prod_a \left(f\left(w_a+[\mathbf{W}\cdot\mathbf{u}]_a\right)\right)^{v_a}\left(1-f\left(w_a+[\mathbf{W}\cdot\mathbf{u}]_a\right)\right)^{1-v_a} \tag{10.43}$$

The parameter list for the recognition model is $\mathcal{W} = (\mathbf{w}, \mathbf{W})$. This distribution is only an approximate inverse of the generative model because it implies that the components of **v** are independent when, in fact, given a particular input **u**, they are conditionally dependent due to the way they interact in equation 10.42 to generate **u** (this is the same assumption as in the mean-field approximate distribution for the Boltzmann machine, except that the parameters of the distribution here are shared between all input cases).

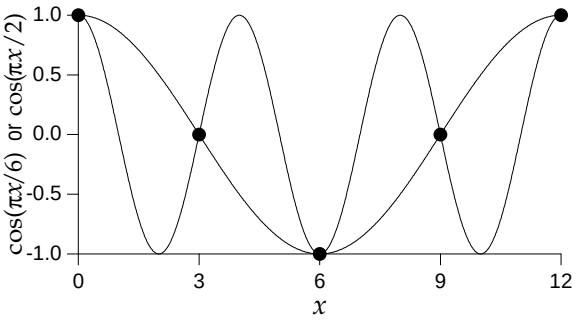

Figure 2.9 Aliasing and the Nyquist frequency. The two curves are the functions $\cos(\pi x/6)$ and $\cos(\pi x/2)$ plotted against x, and the dots show points sampled with a spacing of $\Delta x = 3$. The Nyquist frequency in this case is $\pi/3$, and the two cosine curves match at the sampled points because their spatial frequencies satisfy the relation $2\pi/3 - \pi/6 = \pi/2$.

can be confused with one another in this way, a phenomenon known as aliasing. Conversely, if an image is constructed solely of frequencies less than K_{nyq}, it can be reconstructed perfectly from the finite set of samples provided by the array. There are 120 cones per degree at the fovea of the macaque retina, which makes $K_{\text{nyq}}/(2\pi) = 1/(2\Delta x) = 60$ cycles per degree. In this result, we have divided the right side of equation 2.20, which gives K_{nyq} in units of radians per degree, by 2π to convert the answer to cycles per degree.

2.4 Reverse-Correlation Methods: Simple Cells

The spike-triggered average for visual stimuli is defined, as in chapter 1, as the average over trials of stimuli evaluated at times $t_i - \tau$, where t_i for $i = 1, 2, \ldots, n$ are the spike times. Because the light intensity of a visual image depends on location as well as time, the spike-triggered average stimulus is a function of three variables,

$$C(x, y, \tau) = \frac{1}{\langle n \rangle} \left\langle \sum_{i=1}^{n} s(x, y, t_i - \tau) \right\rangle . \tag{2.21}$$

Here, as in chapter 1, the brackets denote trial averaging, and we have used the approximation $1/n \approx 1/\langle n \rangle$. $C(x, y, \tau)$ is the average value of the visual stimulus at the point (x, y) a time τ before a spike was fired. Similarly, we can define the correlation function between the firing rate at time t and the stimulus at time $t + \tau$, for trials of duration T, as

$$Q_{rs}(x, y, \tau) = \frac{1}{T} \int_0^T dt\, r(t) s(x, y, t + \tau) . \tag{2.22}$$

Appendix), the weight change can be multiplied by $\mathbf{W}^T\mathbf{W}$ without affecting the fixed points of the update rule. This means that the alternative learning rule

$$\mathbf{W}_{ab} \rightarrow \mathbf{W}_{ab} + \epsilon \left(\mathbf{W}_{ab} + g'(v_a)\,[\mathbf{v} \cdot \mathbf{W}]_b\right) \tag{10.39}$$

has the same potential final weight matrices as equation 10.38. This is called a natural gradient rule, and it avoids the matrix inversion of $\mathbf{W}$ as well as providing faster convergence. Equation 10.39 can be interpreted as the sum of an anti-decay term that forces $\mathbf{W}$ away from 0, and a generalized type of anti-Hebbian term. The choice of prior $p[v] \propto 1/\cosh(v)$ makes $g'(v) = -\tanh(v)$ and produces the rule

$$\mathbf{W}_{ab} \rightarrow \mathbf{W}_{ab} + \epsilon \left(\mathbf{W}_{ab} - \tanh(v_a)\,[\mathbf{v} \cdot \mathbf{W}]_b\right) . \tag{10.40}$$

This algorithm is called independent components analysis. Just as the sparse coding network is a nonlinear generalization of factor analysis, independent components analysis is a nonlinear generalization of principal components analysis that attempts to account for non-Gaussian features of the input distribution. The generative model is based on the assumption that $\mathbf{u} = \mathbf{G} \cdot \mathbf{v}$. Some other technical conditions must be satisfied for independent components analysis to extract reasonable causes. Specifically, the prior distributions over causes $p[v] \propto \exp(g(v))$ must be non-Gaussian and, at least to the extent of being correctly super- or sub-Gaussian, must faithfully reflect the actual distribution over causes. The particular form $p[v] \propto 1/\cosh(v)$ is super-Gaussian, and thus generates a sparse prior. There are variants of independent components analysis in which the prior distributions are adaptive.

The independent components algorithm was suggested by Bell and Sejnowski (1995) from the different perspective of maximizing the mutual information between $\mathbf{u}$ and $\mathbf{v}$ when $v_a(\mathbf{u}) = f([\mathbf{W} \cdot \mathbf{u}]_a)$, with a particular, monotonically increasing nonlinear function f. Maximizing the mutual information in this context requires maximizing the entropy of the distribution over $\mathbf{v}$. This, in turn, requires the components of $\mathbf{v}$ to be as independent as possible because redundancy between them reduces the entropy. In the case that $f(v) = g'(v)$, the expression for the entropy is the same as that for the log likelihood in equation 10.37, up to constant factors. Thus, maximizing the entropy and performing maximum likelihood density estimation are identical.

One advantage independent components analysis has over other sparse coding algorithms is that, because the recognition model is an exact inverse of the generative model, receptive as well as projective fields can be constructed. Just as the projective field for v_a can be represented by the matrix elements G_{ab} for all b values, the receptive field is given by W_{ab} for all b.

To illustrate independent components analysis, figure 10.8 shows an (admittedly bizarre) example of its application to the sounds created by tapping a tooth while adjusting the shape of the mouth to reproduce a

Spatial Receptive Fields

Figures 2.10A and C show the spatial structure of spike-triggered average stimuli for two simple cells in the primary visual cortex of a cat (area 17) with approximately separable space-time receptive fields. These receptive fields are elongated in one direction. There are some regions within the receptive field where D_s is positive, called ON regions, and others where it is negative, called OFF regions. The integral of the linear kernel times the stimulus can be visualized by noting how the OFF (black) and ON (white) regions overlap the image (see figure 2.11) . The response of a neuron is enhanced if ON regions are illuminated ($s > 0$) or if OFF regions are darkened ($s < 0$) relative to the background level of illumination. Conversely, they are suppressed by darkening ON regions or illuminating OFF regions. As a result, the neurons of figures 2.10A and C respond most vigorously to light-dark edges positioned along the border between the ON and OFF regions, and oriented parallel to this border and to the elongated direction of the receptive fields (figure 2.11). Figures 2.10 and 2.11 show receptive fields with two major subregions. Simple cells are found with from one to five subregions. Along with the ON-OFF patterns we have seen, another typical arrangement is a three-lobed receptive field with OFF-ON-OFF or ON-OFF-ON subregions.

Gabor function

A mathematical approximation of the spatial receptive field of a simple cell is provided by a Gabor function, which is a product of a Gaussian function and a sinusoidal function. Gabor functions are by no means the only functions used to fit spatial receptive fields of simple cells. For example, gradients of Gaussians are sometimes used. However, we will stick to Gabor functions, and to simplify the notation, we choose the coordinates x and y so that the borders between the ON and OFF regions are parallel to the y axis. We also place the origin of the coordinates at the center of the receptive field. With these choices, we can approximate the observed receptive field structures using the Gabor function

$$D_s(x, y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left(-\frac{x^2}{2\sigma_x^2} - \frac{y^2}{2\sigma_y^2}\right) \cos(kx - \phi) \,. \tag{2.27}$$

rf size σ_x, σ_y

preferred spatial frequency k

preferred spatial phase ϕ

The parameters in this function determine the properties of the spatial receptive field: σ_x and σ_y determine its extent in the x and y directions, respectively; k, the preferred spatial frequency, determines the spacing of light and dark bars that produce the maximum response (the preferred spatial wavelength is $2\pi/k$); and ϕ is the preferred spatial phase, which determines where the ON-OFF boundaries fall within the receptive field. For this spatial receptive field, the sinusoidal grating described by equation 2.18 that produces the maximum response for a fixed value of A has $K = k$, $\Phi = \phi$, and $\Theta = 0$.

Figures 2.10B and 2.10D show Gabor functions chosen specifically to match the data in figures 2.10A and 2.10C. Figure 2.12 shows x and y plots of a variety of Gabor functions with different parameter values. As

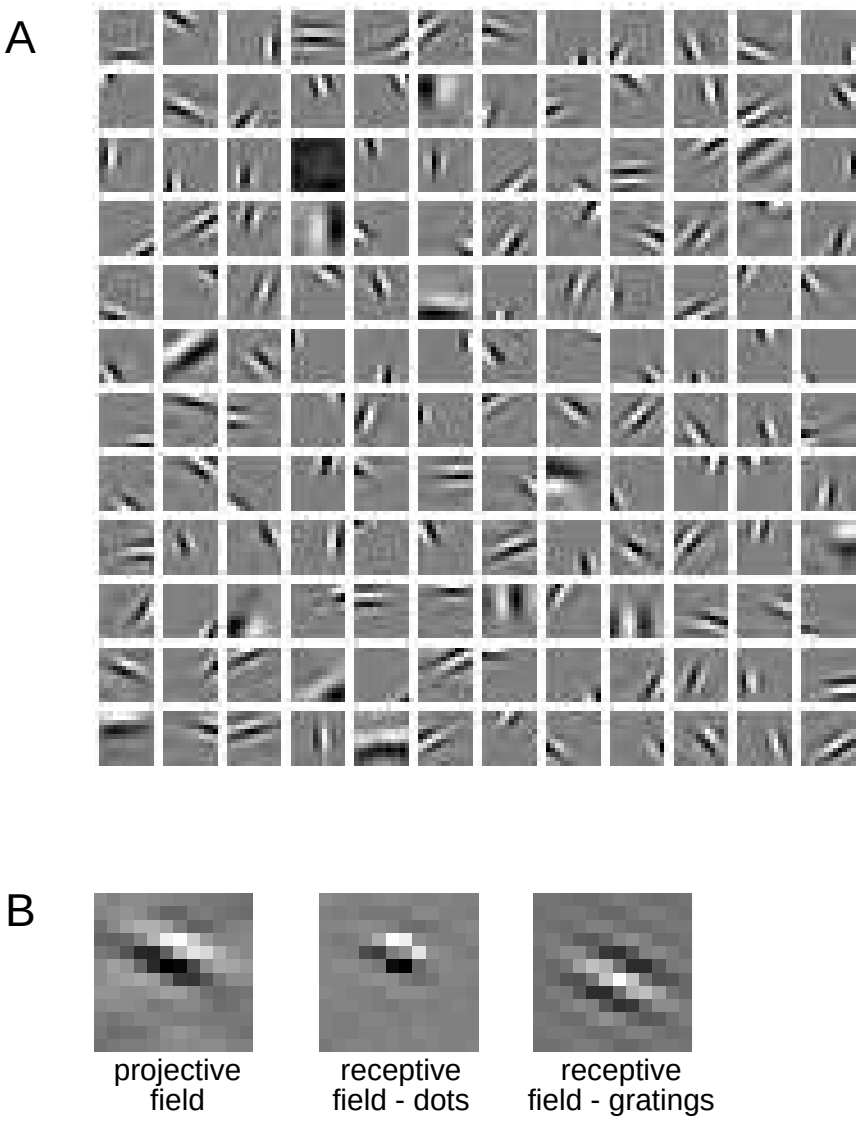

Figure 10.7 Projective and receptive fields for a sparse coding network with $N_u = N_v = 144$. (A) Projective fields $\mathbf{G}_{ab}$ with a indexing representational units (the components of $\mathbf{v}$), and b indexing input units $\mathbf{u}$ on a 12×12 pixel grid. Each box represents a different a value, and the b values are represented within the box by the corresponding input location. Weights are represented by the gray-scale level, with gray indicating 0. (B) The relationship between projective and receptive fields. The left panel shows the projective field of one of the units in A. The middle and right panels show its receptive field mapped using inputs generated by dots and by gratings, respectively. (Adapted from Olshausen & Field, 1997.)

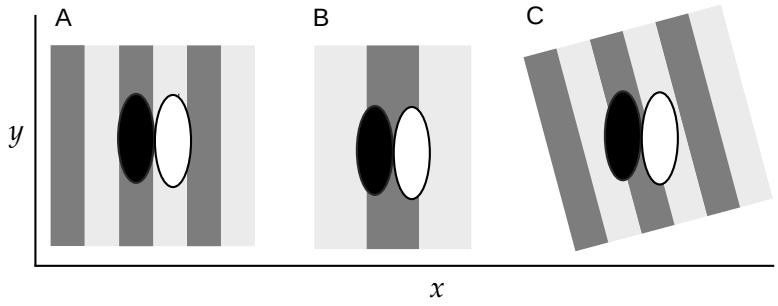

Figure 2.11 Grating stimuli superimposed on spatial receptive fields similar to those shown in figure 2.10. The receptive field is shown as two oval regions, one dark to represent an OFF area where $D_s < 0$ and one white to denote an ON region where $D_s > 0$. (A) A grating with the spatial wavelength, orientation, and spatial phase shown produces a high firing rate because a dark band completely overlaps the OFF area of the receptive field and a light band overlaps the ON area. (B) The grating shown is nonoptimal due to a mismatch in both the spatial phase and frequency, so that the ON and OFF regions each overlap both light and dark stripes. (C) The grating shown is at a nonoptimal orientation because each region of the receptive field overlaps both light and dark stripes.

seen in figure 2.12, Gabor functions can have various types of symmetry, and variable numbers of significant oscillations (or subregions) within the Gaussian envelope. The number of subregions within the receptive field is determined by the product $k\sigma_x$ and is typically expressed in terms of a quantity known as the bandwidth b. The bandwidth is defined as $b = \log_2(K_+/K_-)$, where $K_+ > k$ and $K_- < k$ are the spatial frequencies of gratings that produce one-half the response amplitude of a grating with $K = k$. High bandwidths correspond to low values of $k\sigma_x$, meaning that the receptive field has few subregions and poor spatial frequency selectivity. Neurons with more subfields are more selective to spatial frequency, and they have smaller bandwidths and larger values of $k\sigma_x$.

bandwidth

The bandwidth is the width of the spatial frequency tuning curve measured in octaves. The spatial frequency tuning curve as a function of K for a Gabor receptive field with preferred spatial frequency k and receptive field width σ_x is proportional to $\exp(-\sigma_x^2(k-K)^2/2)$ (see equation 2.34 below). The values of K_+ and K_- needed to compute the bandwidth are thus determined by the condition $\exp(-\sigma_x^2(k-K_\pm)^2/2) = 1/2$. Solving this equation gives $K_\pm = k \pm (2\ln(2))^{1/2}/\sigma_x$, from which we obtain

$$b = \log_2\left(\frac{k\sigma_x + \sqrt{2\ln(2)}}{k\sigma_x - \sqrt{2\ln(2)}}\right) \quad \text{or} \quad k\sigma_x = \sqrt{2\ln(2)}\,\frac{2^b+1}{2^b-1}. \tag{2.28}$$

Bandwidth is defined only if $k\sigma_x > (2\ln(2))^{1/2}$, but this is usually the case. Bandwidths typically range from about 0.5 to 2.5, corresponding to $k\sigma_x$ between 1.7 and 6.9.

The response characterized by equation 2.27 is maximal if light-dark edges are parallel to the y axis, so the preferred orientation angle is 0. An arbitrary preferred orientation, θ, can be created by rotating the coordinates,

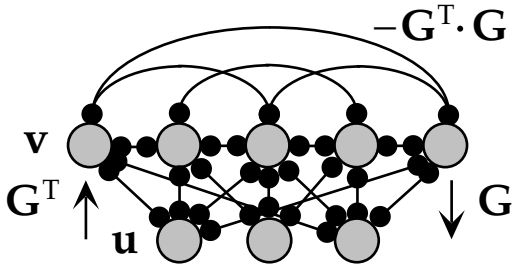

Figure 10.6 A network for sparse coding. This network reproduces equation (10.34), using recurrent weights $-\mathbf{G}^{\mathrm{T}} \cdot \mathbf{G}$ in the $\mathbf{v}$ layer and weights connecting the input units to this layer that are given by the transpose of the matrix $\mathbf{G}$. The reverse connections from the $\mathbf{v}$ layer to the input layer indicate how the mean of the recognition distribution is computed.

be interpreted as a recurrent coupling of the $\mathbf{v}$ units through the matrix $-\mathbf{G}^{\mathrm{T}} \cdot \mathbf{G}$. Finally, the term $g'(v_a)$ plays the same role as the term $-v_a$ that would appear in the rate equations of chapter 7. If $g'(v) \neq -v$, this can be interpreted as a modified form of firing-rate dynamics. Figure 10.6 shows the resulting network. The feedback connections from the $\mathbf{v}$ units to the input units that determine the mean of the generative distribution, $\mathbf{G} \cdot \mathbf{v}$ (equation 10.30), are also shown in this figure.

After $\mathbf{v}(\mathbf{u})$ has been determined during the E phase of EM, a delta rule (chapter 8) is used during the M phase to modify $\mathbf{G}$ and improve the generative model. The full learning rule is given in the appendix. The delta rule follows from maximizing $\mathcal{F}(\mathbf{v}(\mathbf{u}), \mathcal{G})$ with respect to $\mathbf{G}$. A complication arises here because the matrix $\mathbf{G}$ always appears multiplied by $\mathbf{v}$. This means that the bias toward small values of v_a imposed by the prior can be effectively neutralized by scaling up $\mathbf{G}$. This complication results from the approximation of deterministic recognition. To prevent the weights from growing without bound, constraints are applied on the lengths of the generative weights for each cause, $\sum_b G_{ba}^2$, to encourage the variances of all the different v_a to be approximately equal (see the appendix). Further, it is conventional to precondition the inputs before learning by whitening them so that $\langle \mathbf{u} \rangle = 0$ and $\langle \mathbf{u}\mathbf{u} \rangle = \mathbf{I}$. This typically makes learning faster, and it also ensures that the network is forced to find statistical structure beyond second order that would escape simpler methods such as factor analysis or principal components analysis. In the case that the input is created by sampling (e.g., pixelating an image), more sophisticated forms of preconditioning can be used to remove the resulting artifacts.

Applying the sparse coding model to inputs coming from the pixel intensities of small square patches of monochrome photographs of natural scenes leads to selectivities that resemble those of cortical simple cells. Before studying this result, we need to specify how the selectivities of generative models, such as the sparse coding model, are defined. The selectivities of sensory neurons are typically described by receptive fields, as in chapter 2. For a causal model, one definition of a receptive field for unit a is the set of inputs $\mathbf{u}$ for which v_a is likely to take large values. However, it may be

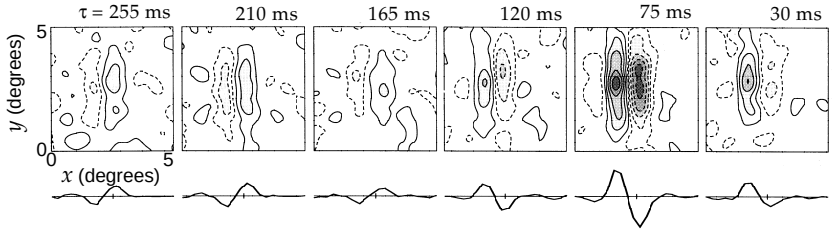

Figure 2.13 Temporal evolution of a spatial receptive field. Each panel is a plot of $D(x, y, \tau)$ for a different value of τ. As in figure 2.10, regions with solid contour curves are areas where $D(x, y, \tau) > 0$ and regions with dashed contours have $D(x, y, \tau) < 0$. The curves below the contour diagrams are one-dimensional plots of the receptive field as a function of x alone. The receptive field is maximally different from 0 for $\tau = 75$ ms with the spatial receptive field reversed from what it was at $\tau = 210$ ms. (Adapted from DeAngelis et al., 1995.)

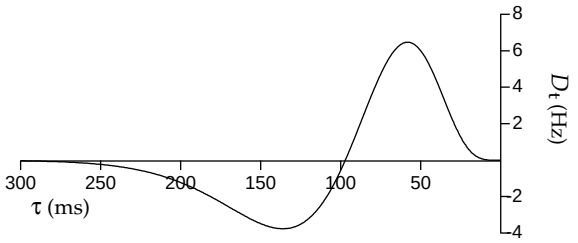

Figure 2.14 Temporal structure of a receptive field. The function $D_t(\tau)$ of equation 2.29 with $\alpha = 1/(15 \text{ ms})$.

returns to 0 as τ increases. Adelson and Bergen (1985) proposed the function shown in figure 2.14,

$$D_t(\tau) = \alpha \exp(-\alpha\tau)\left(\frac{(\alpha\tau)^5}{5!} - \frac{(\alpha\tau)^7}{7!}\right), \tag{2.29}$$

for $\tau \geq 0$, and $D_t(\tau) = 0$ for $\tau < 0$. Here, α is a constant that sets the scale for the temporal development of the function. Single-phase responses are also seen for V1 neurons, and these can be described by eliminating the second term in equation 2.29. Three-phase responses, which are sometimes seen, must be described by a more complicated function.

Response of a Simple Cell to a Counterphase Grating

The response of a simple cell to a counterphase grating stimulus (equation 2.18) can be estimated by computing the function $L(t)$. For the separable receptive field given by the product of the spatial factor in equation 2.27 and the temporal factor in 2.29, the linear estimate of the response can be written as the product of two terms,

$$L(t) = L_s L_t(t), \tag{2.30}$$

More formally, sparseness has been defined in a variety of ways. Sparseness of a distribution is sometimes linked to a high value of a measure called kurtosis. Kurtosis of a distribution $p[v]$ is defined as *kurtosis*

$$k = \frac{\int dv\, p[v](v - \overline{v})^4}{\left(\int dv\, p[v](v - \overline{v})^2\right)^2} - 3 \quad \text{with} \quad \overline{v} = \int dv\, p[v]v\,, \tag{10.29}$$

and it takes the value 0 for a Gaussian distribution. Positive values of k are taken to imply sparse distributions, which are also called super-Gaussian or leptokurtotic. Distributions with $k < 0$ are called sub-Gaussian or platykurtotic. This is a slightly different definition of sparseness from being heavy-tailed.

A sparse representation over a large population of neurons might more naturally be defined as one in which each input is encoded by a small number of the neurons in the population. Unfortunately, identifying this form of sparseness experimentally is difficult.

Sparse coding can arise in generative models that have sparse prior distributions over causes. Unlike factor analysis and principal components analysis, sparse coding does not stress minimizing the number of representing units (i.e., components of $\mathbf{v}$), and sparse representations may require large numbers of units. This is not a disadvantage for modeling the visual system because representations in visual areas are indeed greatly expanded at various steps along the pathway. For example, there are around 40 cells in primary visual cortex for each cell in the visual thalamus. Downstream processing can benefit greatly from sparse representations because, for one thing, they minimize interference between different patterns of input.

Because they employ Gaussian priors, factor analysis and principal components analysis do not generate sparse representations. The mixture of Gaussians model is extremely sparse because each input is represented by a single cause. This may be reasonable for relatively simple input patterns, but for complex stimuli such as images, we seek something between these extremes. Olshausen and Field (1996, 1997) suggested such a model by considering a nonlinear version of factor analysis. In this model, the distribution of $\mathbf{u}$ given $\mathbf{v}$ is Gaussian with a diagonal covariance matrix, as for factor analysis, but the prior distribution over causes is sparse. Defined in terms of a function $g(v)$ (as in figure 10.5), the model has

$$p[\mathbf{v}; \mathcal{G}] \propto \prod_{a=1}^{N_v} \exp(g(v_a)) \quad \text{and} \quad p[\mathbf{u}|\mathbf{v}; \mathcal{G}] = \mathcal{N}(\mathbf{u}; \mathbf{G} \cdot \mathbf{v}, \mathbf{\Sigma})\,. \tag{10.30}$$

The prior $p[\mathbf{v}; \mathcal{G}]$ should be normalized so that its integral over $\mathbf{v}$ is 1, but we omit the normalization factor to simplify the equations.

The prior $p[\mathbf{v}; \mathcal{G}]$ in equation 10.30 makes the components of $\mathbf{v}$ mutually independent because it is a product. If we took $g(v) = -v^2$, $p[\mathbf{v}; \mathcal{G}]$ would be Gaussian (dotted lines in figures 10.5B and 10.5C), and the model would

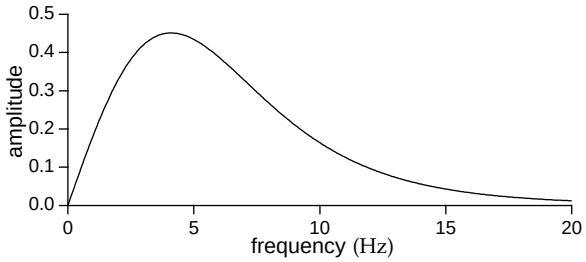

Figure 2.16 Frequency response of a model simple cell based on the temporal kernel of equation 2.29. The amplitude of the sinusoidal oscillations of $L_t(t)$ produced by a counterphase grating is plotted as a function of the temporal oscillation frequency, $\omega/2\pi$.

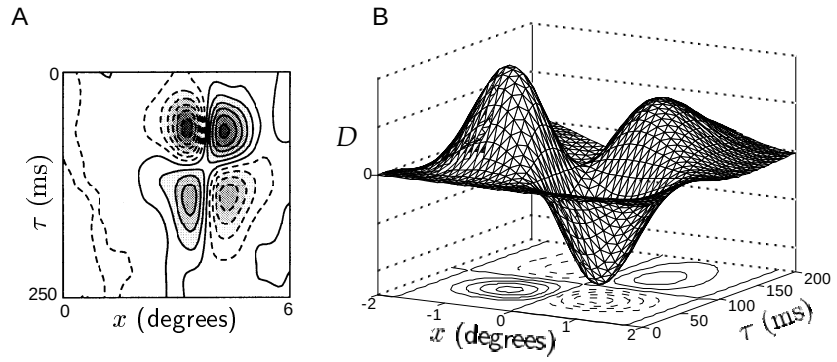

Figure 2.17 A separable space-time receptive field. (A) An x-τ plot of an approximately separable space-time receptive field from cat primary visual cortex. OFF regions are shown with dashed contour lines and ON regions with solid contour lines. The receptive field has side-by-side OFF and ON regions that reverse as a function of τ. (B) Mathematical description of the space-time receptive field in A constructed by multiplying a Gabor function (evaluated at $y = 0$) with $\sigma_x = 1^\circ$, $1/k = 0.56^\circ$, and $\phi = \pi/2$ by the temporal kernel of equation 2.29 with $1/\alpha = 15$ ms. (A adapted from DeAngelis et al., 1995.)

stimulus ($\omega/2\pi$ rather than the angular frequency ω) in figure 2.16. The peak value around 4 Hz and roll-off above 10 Hz are typical for V1 neurons and for cortical neurons in other primary sensory areas as well.

Space-Time Receptive Fields

To display the function $D(x, y, \tau)$ in a space-time plot rather than as a sequence of spatial plots (as in figure 2.13), we suppress the y dependence and plot an x-τ projection of the space-time kernel. Figure 2.17A shows a space-time plot of the receptive field of a simple cell in the cat primary visual cortex. This receptive field is approximately separable, and it has OFF and ON subregions that reverse to ON and OFF subregions as a function of τ, similar to the reversal seen in figure 2.13. Figure 2.17B shows an

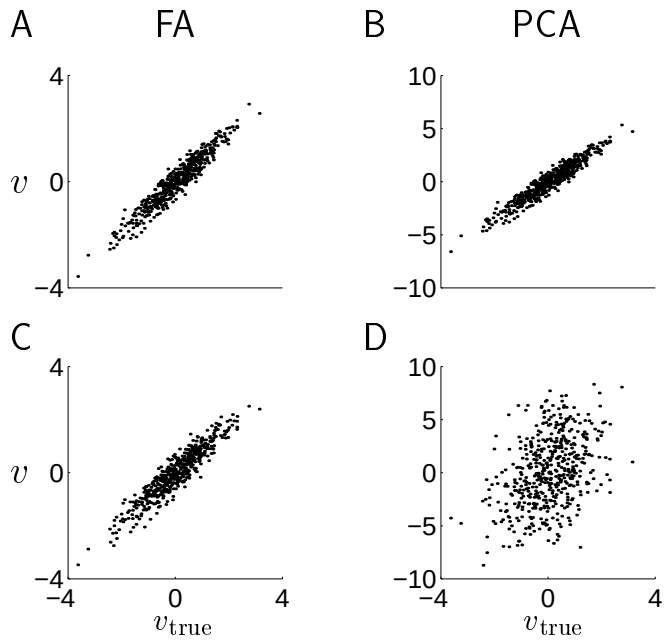

Figure 10.4 Factor analysis (FA) and principal components analysis (PCA) applied to 500 samples of noisy input reflecting a single underlying cause v_{true}. For A and B, $\langle u_i u_j \rangle = 1 + 0.25\delta_{ij}$, whereas for C and D, one sensor is corrupted by independent noise with standard deviation 3 rather than 0.5. The plots compare the values of the true cause v_{true} and the cause v inferred by the model.

best we can expect is for the v values to be well correlated with the values of v_{true}. When the input components are equally variable (figure 10.4A and 10.4B), this is indeed what happens for both factor and principal components analysis. However, when u_3 is much more variable than the other components, principal components analysis (figure 10.4D) is fooled by the extra variance and finds a cause v that does not correlate very well with v_{true}. On the other hand, factor analysis (figure 10.4C) is affected only by the covariance between the input components and not by their individual variances (which are absorbed into $\mathbf{\Sigma}$), so the cause it finds is not significantly perturbed (merely slightly degraded) by the added sensor noise.

In chapter 8, we noted that principal components analysis maximizes the mutual information between the input and output under the assumption of a linear Gaussian model. This property, and the fact that principal components analysis minimizes the reconstruction error of equation 10.27, have themselves been suggested as goals for representational learning. We have now shown how they are also related to density estimation.

Both principal components analysis and factor analysis produce a marginal distribution $p[\mathbf{u}; \mathcal{G}]$ that is Gaussian. If the actual input distribution $p[\mathbf{u}]$ is non-Gaussian, the best that these models can do is to match the mean and covariance of $p[\mathbf{u}]$; they will fail to match higher-order moments. If the input is whitened to increase coding efficiency, as discussed

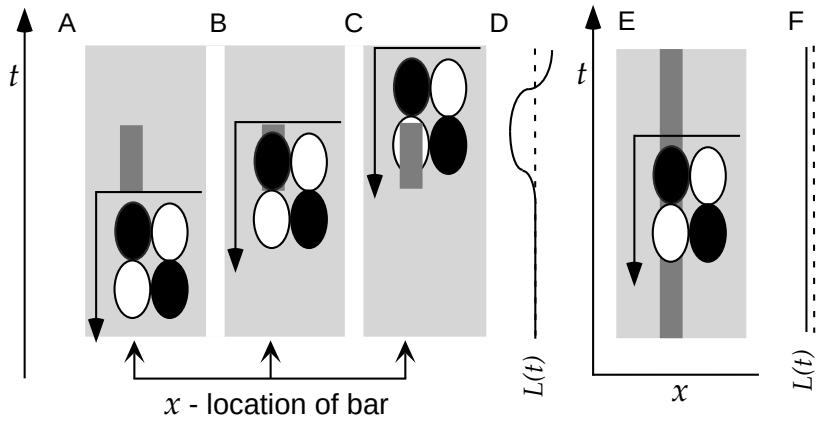

Figure 2.19 Responses to dark bars estimated from a separable space-time receptive field. Dark ovals in the receptive field diagrams are OFF regions and light circles are ON regions. The linear estimate of the response at any time is determined by positioning the receptive field diagram so that its horizontal axis matches the time of response estimation and noting how the OFF and ON regions overlap with the image. (A-C) The image is a dark bar that is flashed on for a short interval of time. There is no response (A) until the dark image overlaps the OFF region (B) when $L(t) > 0$. The response is later suppressed when the dark bar overlaps the ON region (C) and $L(t) < 0$. (D) A plot of $L(t)$ versus time corresponding to the responses generated in A-C. Time runs vertically in this plot, and $L(t)$ is plotted horizontally with the dashed line indicating the zero axis and positive values plotted to the left. (E) The image is a static dark bar. The bar overlaps both an OFF and an ON region, generating opposing positive and negative contributions to $L(t)$. (F) The weak response corresponding to E, plotted as in D.

versed ON region for large τ, generating opposing positive and negative contributions to $L(t)$. The flashed dark bar of figures 2.19A-C is a more effective stimulus because there is a time when it overlaps only the OFF region.

Figure 2.20 shows why a moving grating is a particularly effective stimulus. The grating moves to the left in 2.20A-C. At the time corresponding to the positioning of the receptive field diagram in 2.20A, a dark band stimulus overlaps both OFF regions and light bands overlap both ON regions. Thus, all four regions contribute positive amounts to $L(t)$. As time progresses and the receptive field moves upward in the figure, the alignment will sometimes be optimal, as in 2.20A, and sometimes nonoptimal, as in 2.20B. This produces an $L(t)$ that oscillates as a function of time between positive and negative values (2.20C). Figures 2.20D-F show that a neuron with this receptive field responds equally to a grating moving to the right. Like the left-moving grating in figures 2.20A-C, the right-moving grating can overlap the receptive field in an optimal manner (2.20D), producing a strong response, or in a maximally negative manner (2.20E), producing strong suppression of response, again resulting in an oscillating response (2.20F). Separable space-time receptive fields can produce responses that are maximal for certain speeds of grating motion, but they are not sensitive to the direction of motion.

where expressions for $\mathbf{W}$ and $\mathbf{\Psi}$ are given in the appendix. These do not depend on the input $\mathbf{u}$, so factor analysis involves a linear relation between the input and the mean of the recognition distribution. EM, as applied to an invertible model, can be used to adjust $\mathcal{G} = (\mathbf{G}, \mathbf{\Sigma})$ on the basis of the input data. The resulting learning rules are given in the appendix. For the case of a single cause v, these reduce to equation 10.5.

In this case, we can understand the goal of density estimation in an additional way. By direct calculation, as in equation 10.1, the marginal distribution for $\mathbf{u}$ is

$$p[\mathbf{u}; \mathcal{G}] = \mathcal{N}(\mathbf{u}; \mathbf{0}, \mathbf{G} \cdot \mathbf{G}^{\mathrm{T}} + \mathbf{\Sigma}), \tag{10.23}$$

where $[\mathbf{G}^{\mathrm{T}}]_{ab} = [\mathbf{G}]_{ba}$ and $[\mathbf{G} \cdot \mathbf{G}^{\mathrm{T}}]_{ab} = \sum_c G_{ac} G_{bc}$ (see the Mathematical Appendix). Maximum likelihood density estimation requires determining the $\mathcal{G}$ that makes $\mathbf{G} \cdot \mathbf{G}^{\mathrm{T}} + \mathbf{\Sigma}$ match, as closely as possible, the covariance matrix of the input distribution.

Principal Components Analysis

In the same way that setting the parameters Σ_v to 0 in the mixture of Gaussians model leads to the K-means algorithm, setting all the variances in factor analysis to 0 leads to another well-known method, principal components analysis (which we also discuss in chapter 8). To see this, consider the case of a single factor. This means that v is a single number, and that the mean of the distribution $p[\mathbf{u}|v; \mathcal{G}]$ is $v\mathbf{g}$, where the vector $\mathbf{g}$ replaces the matrix $\mathbf{G}$ of the general case. The elements of the diagonal matrix $\mathbf{\Sigma}$ are set to a single variance Σ, which we shrink to 0.

As $\Sigma \to 0$, the Gaussian distribution $p[\mathbf{u}|v; \mathcal{G}]$ in equation 10.20 approaches a δ function (see the Mathematical Appendix), and it can generate only the single vector $\mathbf{u}(v) = v\mathbf{g}$ from cause v. Similarly, the recognition distribution of equation 10.22 becomes a δ function, making the recognition process deterministic with $v(\mathbf{u}) = \mathbf{W} \cdot \mathbf{u}$ given by the mean of the recognition distribution of equation 10.22. Using the expression for $\mathbf{W}$ in the appendix in the limit $\Sigma \to 0$, we find

$$v(\mathbf{u}) = \frac{\mathbf{g} \cdot \mathbf{u}}{|\mathbf{g}|^2}. \tag{10.24}$$

This is the result of the E phase of EM. In the M phase, we maximize

$$\mathcal{F}(v(\mathbf{u}), \mathcal{G}) = \langle \ln p[v(\mathbf{u}), \mathbf{u}; \mathcal{G}] \rangle = K - \frac{N_u \ln \Sigma}{2} - \left\langle \frac{v^2(\mathbf{u})}{2} + \frac{|\mathbf{u} - \mathbf{g}v(\mathbf{u})|^2}{2\Sigma} \right\rangle \tag{10.25}$$

with respect to $\mathbf{g}$, without changing the expression for $v(\mathbf{u})$. Here, K is a term independent of $\mathbf{g}$ and Σ. In this expression, the only term that depends on $\mathbf{g}$ is proportional to $|\mathbf{u} - \mathbf{g}v(\mathbf{u})|^2$. Minimizing this in the M

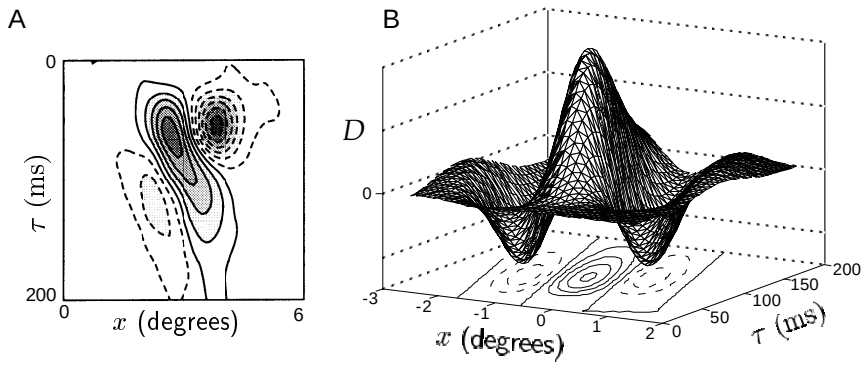

Figure 2.21 A nonseparable space-time receptive field. (A) An x-τ plot of the space-time receptive field of a neuron from cat primary visual cortex. OFF regions are shown with dashed contour lines and ON regions with solid contour lines. The receptive field has a central ON region and two flanking OFF regions that shift to the left over time. (B) Mathematical description of the space-time receptive field in A constructed from equations 2.35 - 2.37. The Gabor function used (evaluated at $y = 0$) had $\sigma_x = 1^\circ$, $1/k = 0.5^\circ$, and $\phi = 0$. D_t is given by the expression in equation 2.29 with $\alpha = 20$ ms, except that the second term, with the seventh power function, was omitted because the receptive field does not reverse sign in this example. The x-τ rotation angle used was $\psi = \pi/9$, and the conversion factor was $c = 0.02\ ^\circ/\text{ms}$. (A adapted from DeAngelis et al., 1995.)

with

$$x' = x\cos(\psi) - c\tau\sin(\psi) \tag{2.36}$$

and

$$\tau' = \tau\cos(\psi) + \frac{x}{c}\sin(\psi)\,. \tag{2.37}$$

The factor c converts between the units of time (ms) and space (degrees), and ψ is the space-time rotation angle. The rotation operation is not the only way to generate nonseparable space-time receptive fields. They are often constructed by adding together two or more separable space-time receptive fields with different spatial and temporal characteristics.

Figure 2.22 shows how a nonseparable space-time receptive field can produce a response that is sensitive to the direction of motion of a grating. Figures 2.22A-C show a left-moving grating and, in 2.22A, the receptive field is positioned at a time when a light area of the image overlaps the central ON region and dark areas overlap the flanking OFF regions. This produces a large positive $L(t)$. At other times, the alignment is nonoptimal (2.22B), and over time, $L(t)$ oscillates between large positive and negative values (2.22C). The nonseparable space-time receptive field does not overlap optimally with the right-moving grating of figures 2.22D-F at any time, and the response is correspondingly weaker (2.22F). Thus, a neuron with a nonseparable space-time receptive field can be selective for the direction of motion of a grating and for its velocity, responding most vigorously to an optimally spaced grating moving at a velocity given, in terms of the parameters in equation 2.36, by $c\tan(\psi)$.

direction selectivity
preferred velocity

10.3 Causal Models for Density Estimation

In this section, we present a number of models in which representational learning is achieved through density estimation. The mixture of Gaussians and factor analysis models that we have already mentioned are examples of invertible generative models with probabilistic recognition. *K*-means is a limiting case of mixture of Gaussians with deterministic recognition, and principal components analysis is a limiting case of factor analysis with deterministic recognition. We consider two other models with deterministic recognition: independent components analysis, which is invertible; and sparse coding, which is noninvertible. Our final example, the Helmholtz machine, is noninvertible with probabilistic recognition. The Boltzmann machine, discussed in chapters 7 and 8, is an additional example that is closely related to the causal models discussed here. We summarize and interpret general properties of representations derived from causal models at the end of the chapter. The table in the appendix summarizes the generative and recognition distributions and the learning rules for all the models we discuss.

Mixture of Gaussians

The model applied in the introduction to the data in figure 10.1A is a mixture of Gaussians model. That example involved two causes and two Gaussian distributions, but we now generalize this to N_v causes, each associated with a separate Gaussian distribution. The model is defined by the probability distributions

$$P[v; \mathcal{G}] = \gamma_v \quad \text{and} \quad p[\mathbf{u}|v; \mathcal{G}] = \mathcal{N}(\mathbf{u}; \mathbf{g}_v, \Sigma_v), \tag{10.17}$$

where v takes N_v values representing the different causes and, for an N_u component input vector,

$$\mathcal{N}(\mathbf{u}; \mathbf{g}, \Sigma) = \frac{1}{(2\pi\Sigma)^{N_u/2}} \exp\left(-\frac{|\mathbf{u}-\mathbf{g}|^2}{2\Sigma}\right) \tag{10.18}$$

is a Gaussian distribution with mean $\mathbf{g}$ and variances for the individual components equal to Σ. The function $\mathcal{F}(Q, \mathcal{G})$ for this model is given by an expression similar to equation 10.14 (with slightly different factors if $N_u \neq 2$), leading to the M-phase learning rules given in the appendix. Once the generative model has been optimized, the recognition distribution is constructed from equation 10.3 as

$$P[v|\mathbf{u}; \mathcal{G}] = \frac{\gamma_v \mathcal{N}(\mathbf{u}; \mathbf{g}_v, \Sigma_v)}{\sum_{v'} \gamma_{v'} \mathcal{N}(\mathbf{u}; \mathbf{g}_{v'}, \Sigma_{v'})}. \tag{10.19}$$

K-Means Algorithm

A special case of mixture of Gaussians can be derived in the limit that the variances of the Gaussians are equal and tend toward 0, $\Sigma_v = \Sigma \to 0$. We

because this reproduces the observed contrast dependence. A number of variants and extensions of this idea have also been considered, including, for example, that the denominator of this expression should include L factors for additional neurons with nearby receptive fields. This can account for the effects of visual stimuli outside the "classical" receptive field. Discussion of these effects is beyond the scope of this chapter.

2.5 Static Nonlinearities: Complex Cells

Recall that neurons in primary visual cortex are characterized as simple or complex. While linear methods, such as spike-triggered averages, are useful for revealing the properties of simple cells, at least to a first approximation, complex cells display features that are fundamentally incompatible with a linear description. The spatial receptive fields of complex cells cannot be divided into separate ON and OFF regions that sum linearly to generate the response. Areas where light and dark images excite the neuron overlap, making it difficult to measure and interpret spike-triggered average stimuli. Nevertheless, like simple cells, complex cells are selective to the spatial frequency and orientation of a grating. However, unlike simple cells, complex cells respond to bars of light or dark no matter where they are placed within the overall receptive field. Likewise, the responses of complex cells to grating stimuli show little dependence on spatial phase. Thus, a complex cell is selective for a particular type of image independent of its exact spatial position within the receptive field. This may represent an early stage in the visual processing that ultimately leads to position-invariant object recognition.

spatial-phase invariance

Complex cells also have temporal response characteristics that distinguish them from simple cells. Complex cell responses to moving gratings are approximately constant, not oscillatory as in figures 2.20 and 2.22. The firing rate of a complex cell responding to a counterphase grating oscillating with frequency ω has both a constant component and an oscillatory component with a frequency of 2ω, a phenomenon known as frequency doubling.

frequency doubling

Even though spike-triggered average stimuli and reverse-correlation functions fail to capture the response properties of complex cells, complex-cell responses can be described, to a first approximation, by a relatively straightforward extension of the reverse-correlation approach. The key observation comes from equation 2.34, which shows how the linear response estimate of a simple cell depends on spatial phase for an optimally oriented grating with K not too small. Consider two such responses, labeled L_1 and L_2, with preferred spatial phases ϕ and $\phi - \pi/2$. Including both the spatial and the temporal response factors, we find, for preferred spatial phase ϕ,

$$L_1 = AB(\omega, K)\cos(\phi - \Phi)\cos(\omega t - \delta)\,, \tag{2.39}$$

likelihood of the data points. However, the EM algorithm for maximizing $\mathcal{F}$ is not exactly the same as likelihood maximization by gradient ascent of $\mathcal{F}$. This is because the function Q is held constant during the M phase while the parameters of the generative model are modified. Although $\mathcal{F}$ is equal to L at the beginning of the M phase, exact equality ceases to be true as soon as the parameters are modified, making $P[v|\mathbf{u}; \mathcal{G}]$ different from Q. $\mathcal{F}$ is equal to $L(\mathcal{G})$ again only after the update of Q during the following E phase. At this point, $L(\mathcal{G})$ must have increased since the last E phase, because $\mathcal{F}$ has increased. This shows that the log likelihood increases monotonically during EM until the process converges.

For the example of figure 10.1, the joint probability over causes and inputs is

$$p[v, \mathbf{u}; \mathcal{G}] = \frac{\gamma_v}{2\pi\Sigma_v} \exp\left(-\frac{|\mathbf{u} - \mathbf{g}_v|^2}{2\Sigma_v}\right), \tag{10.13}$$

and thus

$$\mathcal{F} = \left\langle \sum_v Q[v; \mathbf{u}] \left(\ln\left(\frac{\gamma_v}{2\pi}\right) - \ln \Sigma_v - \frac{|\mathbf{u} - \mathbf{g}_v|^2}{2\Sigma_v} - \ln Q[v; \mathbf{u}] \right) \right\rangle . \tag{10.14}$$

The E phase amounts to computing $P[v|\mathbf{u}; \mathcal{G}]$ from equation 10.3 and setting Q equal to it, as in equation 10.12. The M phase involves maximizing $\mathcal{F}$ with respect to $\mathcal{G}$ for this Q. We leave it as an exercise for the reader to show that maximizing equation 10.14 with respect to the parameters γ_v (taking into account the constraint $\sum_v \gamma_v = 1$), $\mathbf{g}_v$, and Σ_v leads to the rules of equation 10.4. For the example of figure 10.3, the joint probability is

$$p[v, \mathbf{u}; \mathcal{G}] = \frac{\exp(-v^2/2)}{\sqrt{2\pi}} \frac{\exp(-\sum_a (u_a - g_a v)^2 / 2\Sigma_a)}{\sqrt{(2\pi)^3 \Sigma_1 \Sigma_2 \Sigma_3}}, \tag{10.15}$$

from which it is straightforward to calculate the relevant $\mathcal{F}$ function and the associated learning rules of equation 10.5.

Noninvertible Deterministic Models

If the generative model is noninvertible, the E phase of the EM algorithm is more complex than simply setting Q equal to $P[v|\mathbf{u}; \mathcal{G}]$, because it is not practical to compute the recognition distribution exactly. The steps taken during the E phase depend on whether the approximation to the inverse of the model is deterministic or probabilistic, although the basic argument is the same in either case.

Deterministic recognition results in a prediction $v(\mathbf{u})$ of the cause underlying input $\mathbf{u}$. In terms of the function $\mathcal{F}$, this amounts to retaining only the single term $v = v(\mathbf{u})$ in the sum in equation 10.8, and for this single term $Q[v(\mathbf{u}); \mathbf{u}] = 1$. Thus, in this case $\mathcal{F}$ is a functional of the function $v(\mathbf{u})$ and a function of the parameters $\mathcal{G}$ given by

$$\mathcal{F}(Q, \mathcal{G}) = \mathcal{F}(v(\mathbf{u}), \mathcal{G}) = \langle \ln P[v(\mathbf{u}), \mathbf{u}; \mathcal{G}] \rangle . \tag{10.16}$$

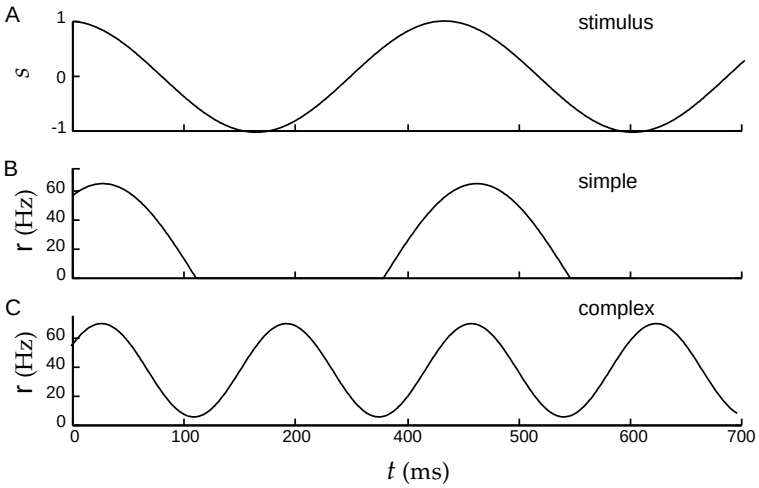

Figure 2.24 Temporal responses of model simple and complex cells to a counterphase grating. (A) The stimulus $s(x, y, t)$ at a given point (x, y) plotted as a function of time. (B) The rectified linear response estimate of a model simple cell to this grating with a temporal kernel given by equation 2.29 with $\alpha = 1/(15 \text{ ms})$. (C) The frequency-doubled response of a model complex cell with the same temporal kernel but with the estimated rate given by a squaring operation rather than rectification. The background firing rate is $r_0 = 5$ Hz. Note the temporal phase shift of both B and C relative to A.

In addition, the last term on the right side of this equation generates the constant component of the complex cell response to a counterphase grating. Figure 2.24 shows a comparison of model simple and complex cell responses to a counterphase grating, and illustrates this phenomenon.

energy model

The description of a complex cell response that we have presented is called an "energy" model because of its resemblance to the equation for the energy of a simple harmonic oscillator. The pair of linear filters used, with preferred spatial phases separated by $\pi/2$, is called a quadrature pair. Because of rectification, the terms L_1^2 and L_2^2 cannot be constructed by squaring the outputs of single simple cells. However, they can each be constructed by summing the squares of rectified outputs from two simple cells with preferred spatial phases separated by π. Thus, we can write the complex cell response as the sum of the squares of four rectified simple cell responses,

$$r(t) = r_0 + G\left([L_1]_+^2 + [L_2]_+^2 + [L_3]_+^2 + [L_4]_+^2\right) , \qquad (2.44)$$

where the different $[L]_+$ terms represent the responses of simple cells with preferred spatial phases ϕ, $\phi + \pi/2$, $\phi + \pi$, and $\phi + 3\pi/2$. While such a construction is possible, it should not be interpreted too literally because complex cells receive input from many sources, including the LGN and other complex cells. Rather, this model should be viewed as purely descriptive. Mechanistic models of complex cells are described at the end of this chapter and in chapter 7.

where K is a term associated with the entropy of the distribution $p[\mathbf{u}]$, that is independent of $\mathcal{G}$. In the second line, we have approximated the integral over all $\mathbf{u}$ values weighted by $p[\mathbf{u}]$ by the average over input data points generated from the distribution $p[\mathbf{u}]$. We assume there are sufficient input data to justify this approximation.

As in the case of the Boltzmann machine discussed in chapter 8, equation 10.6 implies that minimizing the discrepancy between $p[\mathbf{u}]$ and $p[\mathbf{u}; \mathcal{G}]$ amounts to maximizing the log likelihood that the training data could have been created by the generative model,

log likelihood $L(\mathcal{G})$

$$L(\mathcal{G}) = \langle \ln p[\mathbf{u}; \mathcal{G}] \rangle . \tag{10.7}$$

$L(\mathcal{G})$ is the average log likelihood, and the method is known as maximum likelihood density estimation. A theorem due to Shannon describes circumstances under which the generative model that maximizes the likelihood over input data also provides the most efficient way of coding those data, so density estimation is closely related to optimal coding.

maximum likelihood density estimation

Theory of EM

Although stochastic gradient ascent can be used to adjust the parameters of the generative model to maximize the likelihood in equation 10.7 (as it was for the Boltzmann machine), the EM algorithm discussed in the introduction is an alternative procedure that is often more efficient. We applied this algorithm, on intuitive grounds, to the examples of figures 10.1 and 10.3, but we now present a more general and rigorous discussion. This is based on the connection of EM with maximization of the function

$\mathcal{F}(Q, \mathcal{G})$

$$\mathcal{F}(Q, \mathcal{G}) = \left\langle \sum_{v} Q[v; \mathbf{u}] \ln \frac{p[v, \mathbf{u}; \mathcal{G}]}{Q[v; \mathbf{u}]} \right\rangle , \tag{10.8}$$

where $Q[v; \mathbf{u}]$ is any nonnegative function of the discrete argument v and continuous input $\mathbf{u}$ that satisfies

$$\sum_{v} Q[v; \mathbf{u}] = 1 \tag{10.9}$$

for all $\mathbf{u}$. Although, in principle, $Q[v; \mathbf{u}]$ can be any function, we consider it to be an approximate recognition distribution, that is $Q[v; \mathbf{u}] \approx P[v|\mathbf{u}; \mathcal{G}]$.

$\mathcal{F}$ is a useful quantity because, by a rearrangement of terms, it can be written as the difference of the average log likelihood and the average Kullback-Leibler divergence between $Q[v; \mathbf{u}]$ and $P[v|\mathbf{u}; \mathcal{G}]$. This is done by noting that the joint distribution over inputs and causes satisfies $p[v, \mathbf{u}; \mathcal{G}] = P[v|\mathbf{u}; \mathcal{G}]p[\mathbf{u}; \mathcal{G}]$, in addition to 10.2, and using 10.9 and the

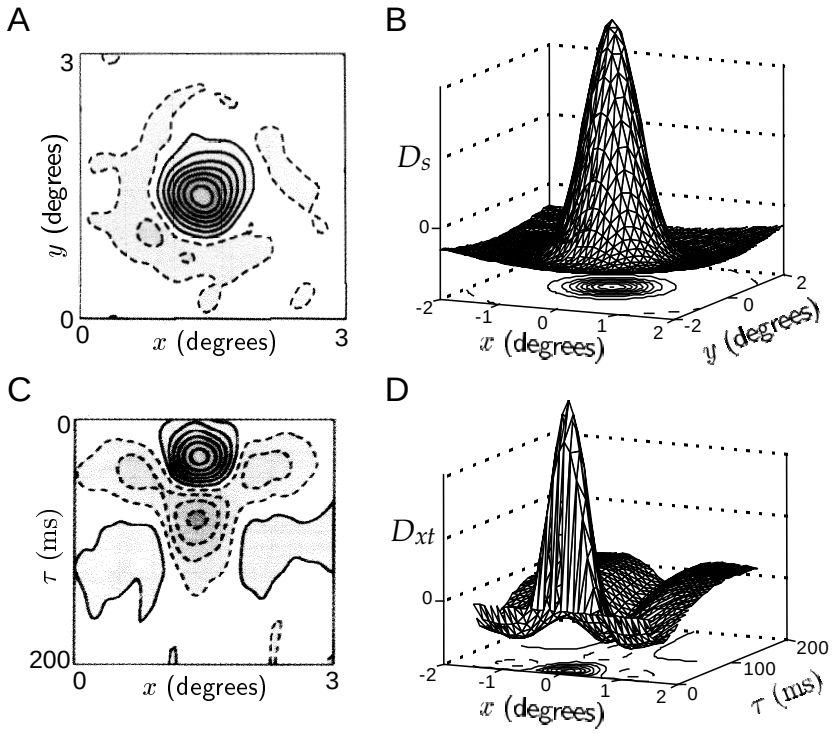

Figure 2.25 Receptive fields of LGN neurons. (A) The center-surround spatial structure of the receptive field of a cat LGN X cell. This has a central ON region (solid contours) and a surrounding OFF region (dashed contours). (B) A fit of the receptive field shown in A using a difference-of-Gaussians function (equation 2.45) with $\sigma_{\text{cen}} = 0.3^\circ$, $\sigma_{\text{sur}} = 1.5^\circ$, and $B = 5$. (C) The space-time receptive field of a cat LGN X cell. Note that the center and surround regions both reverse sign as a function of τ and that the temporal evolution is slower for the surround than for the center. (D) A fit of the space-time receptive field in C using equation 2.46 with the same parameters for the Gaussian functions as in B, and temporal factors given by equation 2.47 with $1/\alpha_{\text{cen}} = 16$ ms for the center, $1/\alpha_{\text{sur}} = 32$ ms for the surround, and $1/\beta_{\text{cen}} = 1/\beta_{\text{sur}} = 64$ ms. (A and C adapted from DeAngelis et al., 1995.)

Separate functions of time multiply the center and surround, but they can both be described by the same functions, using two sets of parameters,

$$D_{\text{t}}^{\text{cen,sur}}(\tau) = \alpha_{\text{cen,sur}}^2 \tau \exp(-\alpha_{\text{cen,sur}}\tau) - \beta_{\text{cen,sur}}^2 \tau \exp(-\beta_{\text{cen,sur}}\tau)\,. \quad (2.47)$$

The parameters α_{cen} and α_{sur} control the latency of the response in the center and surround regions, respectively, and β_{cen} and β_{sur} affect the time of the reversal. This function has characteristics similar to the function in equation 2.29, but the latency effect is less pronounced. Figure 2.25D shows the space-time receptive field of equation 2.46 with parameters chosen to match figure 2.25C.

Figure 2.26 shows the results of a direct test of a reverse-correlation model of an LGN neuron. The kernel needed to describe a particular LGN cell was first extracted by using a white-noise stimulus. This, together with

model, along with a solid line indicating the direction of $\mathbf{g}$. As in figure 10.1B, although the generative model has the capacity to create a data distribution like that in figure 10.3A, the parameters underlying figure 10.3B are clearly inappropriate, and must be adjusted by a learning procedure. Figure 10.3C shows synthetic data after learning, indicating the close match between the marginal distribution $p[\mathbf{u}; \mathcal{G}]$ from the model and the input distribution $p[\mathbf{u}]$.

This model is a simple case of factor analysis; the general case is discussed in section 10.3. The EM algorithm for factor analysis is similar in structure to that for clustering. As before, the basic idea is that if we knew the value of the cause that underlies each input point, we could find the parameters $\mathcal{G}$ easily. Here, the parameters would be determined by solving the linear regression problem that fits the observed inputs to the variable v. This mirrors the observation in our first example that if we knew the cluster assignment for each input point, we could easily find the optimal means and variances of the clusters. Of course, we do not know the values of the causes. Rather, as before, in the E phase of the EM algorithm, the distribution over causes $p[v|\mathbf{u}; \mathcal{G}]$ is calculated from the continuous analog of equation 10.3 ($P[v; \mathcal{G}]$ on the right side of equation 10.3 is replaced by $p[v; \mathcal{G}]$), and this is used as our best current estimate of how likely cause v is associated with input $\mathbf{u}$. Then the M phase consists of weighted linear regression, fitting the observations $\mathbf{u}$ to the variables v weighted by the current recognition probabilities. The result is analogous to equation 10.4; we set

$$g_a = \frac{\langle \int dv\, p[v|\mathbf{u}; \mathcal{G}] v u_a \rangle}{\langle \int dv\, p[v|\mathbf{u}; \mathcal{G}] v^2 \rangle} \quad \text{and} \quad \Sigma_a = \left\langle \int dv\, p[v|\mathbf{u}; \mathcal{G}] (u_a - v g_a)^2 \right\rangle . \tag{10.5}$$

Approximate Recognition

In the two examples we have considered, equation 10.3 was used to obtain the recognition distribution directly from the generative model. For some models, however, it is impractically difficult to evaluate the right side of equation 10.3 and obtain the recognition distribution in this way. We call models in which the recognition distribution can be computed tractably from equation 10.3 invertible, and those in which it cannot be computed tractably, noninvertible. In the latter case, because equation 10.3 cannot be used, recognition is based on an approximate recognition distribution. This is a function $Q[v; \mathbf{u}]$ that approximates the exact recognition distribution $P[v|\mathbf{u}; \mathcal{G}]$. Often, as we discuss in the next section, the best approximation of the recognition distribution comes from adjusting parameters through an optimization procedure. Once this is done, $Q[v; \mathbf{u}]$ provides the model's estimate of the probability that input $\mathbf{u}$ is associated with cause v, and substitutes for the exact recognition distribution $P[v|\mathbf{u}; \mathcal{G}]$.

invertible and noninvertible models

approximate recognition distribution $Q[v; \mathbf{u}]$

The E phase of the EM algorithm in a noninvertible model consists of

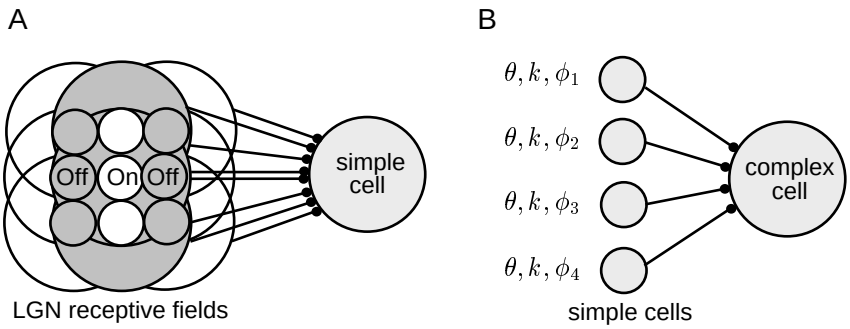

Figure 2.27 (A) The Hubel-Wiesel model of orientation selectivity. The spatial arrangement of the receptive fields of nine LGN neurons are shown, with a row of three ON-center fields flanked on either side by rows of three OFF-center fields. White areas denote ON fields and gray areas, OFF fields. In the model, the converging LGN inputs are summed by the simple cell. This arrangement produces a receptive field oriented in the vertical direction. (B) The Hubel-Wiesel model of a complex cell. Inputs from a number of simple cells with similar orientation and spatial frequency preferences (θ and k), but different spatial phase preferences (ϕ_1, ϕ_2, ϕ_3, and ϕ_4), converge on a complex cell and are summed. This produces a complex cell output that is selective for orientation and spatial frequency, but not for spatial phase. The figure shows four simple cells converging on a complex cell, but additional simple cells can be included to give a more complete coverage of spatial phase.

rangement of LGN receptive fields that, when summed, form bands of ON and OFF regions resembling the receptive field of an oriented simple cell. This model accounts for the selectivity of a simple cell purely on the basis of feedforward input from the LGN. We leave the study of this model as an exercise for the reader. Other models, which we discuss in chapter 7, include the effects of recurrent intracortical connections as well.

Hubel-Wiesel complex cell model

In a previous section, we showed how the properties of complex cell responses could be accounted for by using a squaring static nonlinearity. While this provides a good description of complex cells, there is little indication that complex cells actually square their inputs. Models of complex cells can be constructed without introducing a squaring nonlinearity. One such example is another model proposed by Hubel and Wiesel (1962), which is depicted in figure 2.27B. Here the phase-invariant response of a complex cell is produced by summing together the responses of several simple cells with similar orientation and spatial frequency tuning, but different preferred spatial phases. In this model, the complex cell inherits its orientation and spatial frequency preference from the simple cells that drive it, but spatial phase selectivity is reduced because the outputs of simple cells with a variety of spatial phase selectivities are summed. Analysis of this model is left as an exercise. While the model generates complex cell responses, there are indications that complex cells in primary visual cortex are not driven exclusively by simple cell input. An alternative model is considered in chapter 7.

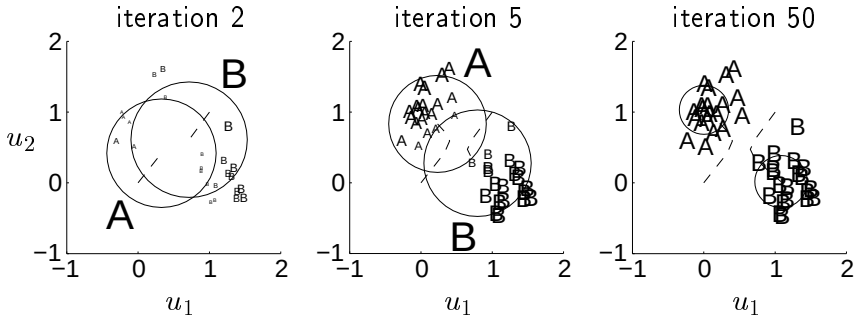

Figure 10.2 EM for clustering. Three stages during the course of EM learning of a generative model. The circles show the Gaussian distributions for clusters A and B (labeled with the largest A and B) as in figure 10.1B and 10.1C. The "trails" behind the centers of the circles plot the change in the mean since the last iteration. The data from figure 10.1A are plotted using the small labels. Label A is used if $P[v = \mathrm{A}|\mathbf{u}; \mathcal{G}] > 0.5$ (and otherwise label B), with the font size proportional to $|P[v = \mathrm{A}|\mathbf{u}; \mathcal{G}] - 0.5|$. This makes the fonts small in regions where the two distributions overlap, even inside one of the circles. The assignment of labels for the two Gaussians (i.e., which is A and which is B) depends on initial conditions.

cluster using the recognition distribution computed from equation 10.3. In other words, the recognition distribution $P[v|\mathbf{u}; \mathcal{G}]$ provides us with our best current guess about the cluster assignment, and this can be used in place of the actual knowledge about which neuron produced which spike. $P[v|\mathbf{u}; \mathcal{G}]$ is thus used to assign the data point $\mathbf{u}$ to cluster v in a probabilistic manner. In this context, $P[v|\mathbf{u}; \mathcal{G}]$ is also called the responsibility of v for $\mathbf{u}$.

responsibility

Following this reasoning, the mean and variance of the Gaussian distribution corresponding to cause v are set equal to a weighted mean and variance of all the data points, with the weight for point $\mathbf{u}$ equal to the current estimate $P[v|\mathbf{u}; \mathcal{G}]$ of the probability that it belongs to cluster v. A similar argument is applied to the mixing proportions, resulting in the equations

$$\gamma_v = \langle P[v|\mathbf{u}; \mathcal{G}] \rangle , \quad \mathbf{g}_v = \frac{\langle P[v|\mathbf{u}; \mathcal{G}]\mathbf{u} \rangle}{\gamma_v} , \quad \Sigma_v = \frac{\langle P[v|\mathbf{u}; \mathcal{G}]|\mathbf{u} - \mathbf{g}_v|^2 \rangle}{2\gamma_v} . \tag{10.4}$$

The angle brackets indicate averages over all the input data points. The factors of γ_v dividing the last two expressions correct for the fact that the number of points in cluster v is expected to be γ_v times the total number of input data points, whereas the full averages denoted by the brackets involve dividing by the total number of data points.

The full EM algorithm consists of two phases that are applied in alternation. In the E (or expectation) phase, the responsibilities $P[v|\mathbf{u}; \mathcal{G}]$ are calculated from equation 10.3. In the M (or maximization) phase, the generative parameters $\mathcal{G}$ are modified according to equation 10.4. The process of determining the responsibilities and then averaging according to them repeats because the responsibilities change when $\mathcal{G}$ is modified. Figure 10.2

E phase
M phase

Rearranging and simplifying this expression gives the condition

$$\Delta t \sum_{k=0}^{\infty} D_k \left(\frac{\Delta t}{T} \sum_{i=0}^{T/\Delta t} s_{i-k} s_{i-j} \right) = \frac{\Delta t}{T} \sum_{i=0}^{T/\Delta t} (\mathsf{r}_i - \mathsf{r}_0)\, s_{i-j} \,. \tag{2.51}$$

If we take the limit $\Delta t \to 0$ and make the replacements $i\Delta t \to t$, $j\Delta t \to \tau$, and $k\Delta t \to \tau'$, the sums in equation 2.51 turn back into integrals, the indexed variables become functions, and we find

$$\int_0^\infty d\tau'\, D(\tau') \left(\frac{1}{T} \int_0^T dt\, s(t-\tau') s(t-\tau) \right) = \frac{1}{T} \int_0^T dt\ (\mathsf{r}(t) - \mathsf{r}_0)\, s(t-\tau) \,. \tag{2.52}$$

The term proportional to r_0 on the right side of this equation can be dropped because the time integral of s is 0. The remaining term is the firing rate-stimulus correlation function evaluated at $-\tau$, $Q_{\mathrm{rs}}(-\tau)$. The term in large parentheses on the left side of 2.52 is the stimulus autocorrelation function. By shifting the integration variable $t \to t + \tau$, we find that it is $Q_{ss}(\tau - \tau')$, so 2.52 can be re-expressed in the form of equation 2.4.

Equation 2.6 provides the solution to equation 2.4 only for a white-noise stimulus. For an arbitrary stimulus, equation 2.4 can easily be solved by the method of Fourier transforms if we ignore causality and allow the estimated rate at time t to depend on the stimulus at times later than t, so that

$$\mathsf{r}_{\mathrm{est}}(t) = \mathsf{r}_0 + \int_{-\infty}^{\infty} d\tau\, D(\tau) s(t-\tau) \,. \tag{2.53}$$

The estimate written in this acausal form satisfies a slightly modified version of equation 2.4,

$$\int_{-\infty}^{\infty} d\tau'\, Q_{ss}(\tau - \tau') D(\tau') = Q_{\mathrm{rs}}(-\tau) \,. \tag{2.54}$$

We define the Fourier transforms (see the Mathematical Appendix)

$$\tilde{D}(\omega) = \int_{-\infty}^{\infty} dt\, D(t) \exp(i\omega t) \ \text{ and } \ \tilde{Q}_{ss}(\omega) = \int_{-\infty}^{\infty} d\tau\, Q_{ss}(\tau) \exp(i\omega\tau) \,, \tag{2.55}$$

as well as $\tilde{Q}_{\mathrm{rs}}(\omega)$ defined analogously to $\tilde{Q}_{ss}(\omega)$.

Equation 2.54 is solved by taking the Fourier transform of both sides. The integral of the product of two functions that appears on the left side of equation 2.54 is called a convolution. To evaluate its Fourier transform, we make use of an important theorem stating that the Fourier transform of a convolution is the product of the Fourier transforms of the two functions involved (see the Mathematical Appendix),

$$\int_{-\infty}^{\infty} d\tau\ \exp(i\omega\tau) \int_{-\infty}^{\infty} d\tau'\, Q_{ss}(\tau - \tau') D(\tau') = \tilde{D}(\omega) \tilde{Q}_{ss}(\omega) \,. \tag{2.56}$$

figure 10.1C, synthetic data points generated by the model (crosses) overlap well with the actual data points seen in figure 10.1A.

The distribution of synthetic data points in figures 10.1B and 10.1C is described by the probability density $p[\mathbf{u}; \mathcal{G}]$ that the generative model synthesizes an input with the value $\mathbf{u}$. This can be computed from the conditional density $p[\mathbf{u}|v; \mathcal{G}]$ and the prior distribution $P[v; \mathcal{G}]$ that define the generative model,

marginal distribution $p[\mathbf{u}; \mathcal{G}]$

$$p[\mathbf{u}; \mathcal{G}] = \sum_v p[\mathbf{u}|v; \mathcal{G}] P[v; \mathcal{G}] . \tag{10.1}$$

The process of summing over all causes is called marginalization, and $p[\mathbf{u}; \mathcal{G}]$ is called the marginal distribution over $\mathbf{u}$. As in chapter 8, we use the additional argument $\mathcal{G}$ to distinguish the distribution of synthetic inputs produced by the generative model, $p[\mathbf{u}; \mathcal{G}]$, from the distribution of actual inputs, $p[\mathbf{u}]$. The process of adjusting the parameters $\mathcal{G}$ to make the distributions of synthetic and real input data points match corresponds to making the marginal distribution $p[\mathbf{u}; \mathcal{G}]$ approximate, as closely as possible, the distribution $p[\mathbf{u}]$ from which the input data points are drawn.

In a later section, we make use of an addition probability distribution associated with the generative model, the joint probability distribution over both causes and inputs, define by

joint distribution $p[v, \mathbf{u}; \mathcal{G}]$

$$p[v, \mathbf{u}; \mathcal{G}] = p[\mathbf{u}|v; \mathcal{G}] P[v; \mathcal{G}] . \tag{10.2}$$

This describes the probability of cause v and input $\mathbf{u}$ both being produced by the generative model.

As mentioned previously, the choice of a particular structure for a generative model reflects our notions and prejudices (i.e., our heuristics) concerning the nature of the causes that underlie input data. Usually, the heuristics consist of biases toward certain types of representations, which are imposed through the choice of the prior distribution $p[v; \mathcal{G}]$. For example, we may want the identified causes to be mutually independent (which leads to a factorial representation or code) or sparse, or of lower dimension than the input data. Many heuristics can be formalized using the information theoretic ideas we discuss in chapter 4.

factorial coding
sparse coding
dimensionality reduction

Recognition Models

Once the optimal generative model has been constructed, the culmination of representational learning is recognition, in which new input data are interpreted in terms of the causes established by the generative model. In probabilistic recognition models, this amounts to determining the probability that cause v is associated with input $\mathbf{u}$, $P[v|\mathbf{u}; \mathcal{G}]$, which is called the posterior distribution over causes or the recognition distribution.

recognition distribution $P[v|\mathbf{u}; \mathcal{G}]$

In the model of figure 10.1, and in many of the models discussed in this chapter, recognition falls directly out of the generative model. The probability of cause v, given input $\mathbf{u}$, $P[v|\mathbf{u}; \mathcal{G}]$, is the statistical inverse of the

optimal) when nonlinearities are present. The self-consistency condition is that when the nonlinear estimate $r_{est} = r_0 + F(L(t))$ is substituted into equation 2.6, the relationship between the linear kernel and the firing rate-stimulus correlation function should still hold. In other words, we require that

$$D(\tau) = \frac{1}{\sigma_s^2 T}\int_0^T dt\, r_{est}(t)s(t-\tau) = \frac{1}{\sigma_s^2 T}\int_0^T dt\, F(L(t))s(t-\tau)\,. \quad (2.63)$$

We have dropped the r_0 term because the time integral of s is 0. In general, equation 2.63 does not hold, but if the stimulus used to extract D is Gaussian white noise, equation 2.63 reduces to a simple normalization condition on the function F. This result is based on the identity, valid for a Gaussian white-noise stimulus,

$$\frac{1}{\sigma_s^2 T}\int_0^T dt\, F(L(t))s(t-\tau) = \frac{D(\tau)}{T}\int_0^T dt\, \frac{dF(L(t))}{dL}\,. \quad (2.64)$$

For the right side of this equation to be $D(\tau)$, the remaining expression, involving the integral of the derivative of F, must be equal to 1. This can be achieved by appropriate scaling of F. The critical identity 2.64 is based on integration by parts for a Gaussian weighted integral. A simplified proof is presented as a problem on the exercise web site.

2.10 Annotated Bibliography

Marmarelis & Marmarelis (1978), **Rieke et al. (1997)**, and **Gabbiani & Koch (1998)** provide general discussions of reverse-correlation methods. A useful reference relevant to our presentation of their application to the visual system is **Carandini et al. (1996)**. Volterra and Wiener functional expansions are discussed in **Wiener (1958)** and **Marmarelis & Marmarelis (1978)**.

General introductions to the visual system include **Hubel & Wiesel (1962, 1977)**, **Orban (1984)**, **Hubel (1988)**, **Wandell (1995)**, and **De Valois & De Valois (1990)**. Our treatment follows **Dowling (1987)** on processing in the retina, and Schwartz (1977), Van Essen et al. (1984), and Rovamo & Virsu (1984) on aspects of the retinotopic map from the eye to the brain. Properties of this map are used to account for aspects of visual hallucinations in Ermentrout & Cowan (1979). We also follow **Movshon et al. (1978a, 1978b)** for definitions of simple and complex cells, Daugman (1985) and Jones & Palmer (1987b) on the use of Gabor functions (Gabor, 1946) to describe visual receptive fields, and **DeAngelis et al. (1995)** on space-time receptive fields. Our description of the energy model of complex cells is based on Adelson & Bergen (1985), which is related to work by Pollen & Ronner (1982), Van Santen & Sperling (1984), and Watson & Ahumada (1985), and to earlier ideas of Reichardt (1961) and Barlow & Levick (1965). Heeger's (1992, 1993) model of contrast saturation is reviewed in **Carandini et al.**

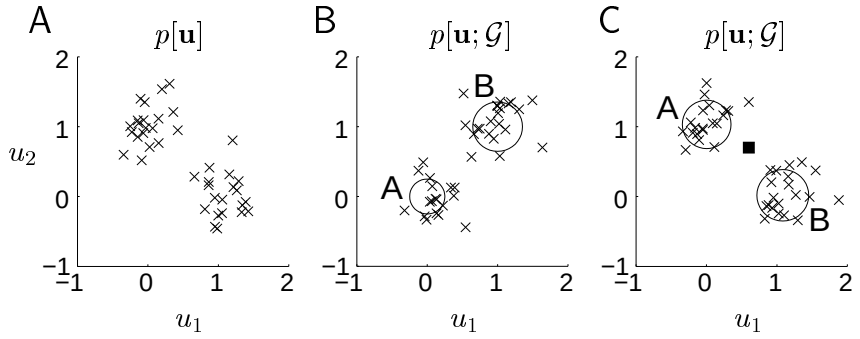

Figure 10.1 Clustering. (A) Input data points drawn from the distribution $p[\mathbf{u}]$ are indicated by the crosses. (B) Initialization for a generative model. The means and twice the standard deviations of the two Gaussians are indicated by the locations and radii of the circles. The crosses show synthetic data, which are samples from the distribution $p[\mathbf{u}; \mathcal{G}]$ of the generative model. (C) Means, standard deviations, and synthetic data points generated by the optimal generative model. The square indicates a new input point that is assigned to either cluster A or cluster B with probabilities computed from the recognition model.

Causal Models

Figure 10.1A provides a simple example of structured data that suggests underlying causes. In this case, the input takes the form of a two-component vector, $\mathbf{u} = (u_1, u_2)$. A collection of sample inputs that we wish to represent in terms of underlying causes is indicated by the 40 crosses in figure 10.1A. These inputs are drawn from a probability density $p[\mathbf{u}]$ that we call the input distribution. Clearly, there are two clusters of points in figure 10.1A, one centered near $(0, 1)$ and the other near $(1, 0)$.

input distribution $p[\mathbf{u}]$

Many processes can generate clustered data. For example, u_1 and u_2 might be characterizations of the voltage recorded on an extracellular electrode in response to an action potential. Interpreted in this way, these data suggest that we are looking at spikes produced by two neurons (called A and B), which are the underlying causes of the two clusters seen in figure 10.1A. A more compact and causal description of the data can be provided by a single output variable v that takes the value A or B for each data point, representing which of the two neurons was responsible for a particular action potential. Directly reading the output of such a model would be an example of deterministic recognition, with $v(\mathbf{u})$ = A or B providing the model's estimate of which neuron produced the spike associated with input $\mathbf{u}$. We consider, instead, a model with probabilistic recognition that estimates the probability that the spike with input data $\mathbf{u}$ was generated by either neuron A or neuron B.

In this example, we assume from the start that there are two possible, mutually exclusive causes for the data points, the two neurons A and B. By making this assumption, which is part of the heuristics underlying the generative model, we avoid the problem of identifying the number of pos-

10 Representational Learning

10.1 Introduction

The response selectivities of individual neurons, and the way they are distributed across neuronal populations, define how sensory information is represented by neural activity. Sensory information is typically represented in multiple brain regions, the visual system being a prime example, with the nature of the representation shifting progressively along the sensory pathway. In previous chapters, we discussed how such representations can be generated by neural circuitry and developed by activity-dependent plasticity. In this chapter, we study neural representations from a computational perspective, asking what goals are served by particular representations, and how appropriate representations might be developed on the basis of input statistics.

re-representation

Constructing new representations of, or re-representing, sensory input is important because sensory receptors often deliver information in a form that is unsuitable for higher-level cognitive tasks. For example, roughly 10^8 photoreceptors provide a pixelated description of the images that appear on our retinas. A list of the membrane potentials of each of these photoreceptors provides a bulky and awkward representation of the visual world, from which it would be difficult to identify directly the face of a friend or a familiar object. Instead, the information provided by photoreceptor outputs is processed in a series of stages involving increasingly sophisticated representations of the visual world. In this chapter, we consider how these more complex and useful representations can be constructed.

generative model

The key to building useful representations for vision lies in determining the structure of visual images and capturing the constraints imposed on images by the natural world. The set of possible pixelated activities arising from natural scenes is richly structured and highly constrained, because images are not generated randomly, but arise from well-defined objects, such as rocks, trees, and people. We call these objects the "causes" of the images. In representational learning, we seek to identify causes by analyzing the statistical structure of visual images and building a model, called the generative model, that is able to reproduce this structure. Iden-

prior probability

- $P[s]$, the probability of stimulus s being presented, often called the prior probability
- $P[\mathbf{r}]$, the probability of response $\mathbf{r}$ being recorded

joint probability

- $P[\mathbf{r}, s]$, the probability of stimulus s being presented and response $\mathbf{r}$ being recorded. This is called the joint probability
- $P[\mathbf{r}|s]$, the conditional probability of evoking response $\mathbf{r}$, given that stimulus s was presented
- $P[s|\mathbf{r}]$, the conditional probability that stimulus s was presented, given that response $\mathbf{r}$ was recorded.

Note that $P[\mathbf{r}|s]$ is the probability of observing the rates $\mathbf{r}$, given that the stimulus took the value s, while $P[\mathbf{r}]$ is the probability of the rates taking the values $\mathbf{r}$ independent of what stimulus was used. $P[\mathbf{r}]$ can be computed from $P[\mathbf{r}|s]$ by summing over all stimulus values weighted by their probabilities,

$$P[\mathbf{r}] = \sum_s P[\mathbf{r}|s]P[s] \quad \text{and similarly} \quad P[s] = \sum_{\mathbf{r}} P[s|\mathbf{r}]P[\mathbf{r}]\,. \tag{3.1}$$

An additional relationship between the probabilities listed above can be derived by noticing that the joint probability $P[\mathbf{r}, s]$ can be expressed as either the conditional probability $P[\mathbf{r}|s]$ times the probability of the stimulus, or as $P[s|\mathbf{r}]$ times the probability of the response,

$$P[\mathbf{r}, s] = P[\mathbf{r}|s]P[s] = P[s|\mathbf{r}]P[\mathbf{r}]\,. \tag{3.2}$$

Bayes theorem

This is the basis of Bayes theorem relating $P[s|\mathbf{r}]$ to $P[\mathbf{r}|s]$:

$$P[s|\mathbf{r}] = \frac{P[\mathbf{r}|s]P[s]}{P[\mathbf{r}]}\,, \tag{3.3}$$

assuming that $P[\mathbf{r}] \neq 0$. Encoding is characterized by the set of probabilities $P[\mathbf{r}|s]$ for all stimuli and responses. Decoding a response, on the other hand, amounts to determining the probabilities $P[s|\mathbf{r}]$. According to Bayes theorem, $P[s|\mathbf{r}]$ can be obtained from $P[\mathbf{r}|s]$, but the stimulus probability $P[s]$ is also needed. As a result, decoding requires knowledge of the statistical properties of experimentally or naturally occurring stimuli.

In the above discussion, we have assumed that both the stimulus and the response are characterized by discrete values so that ordinary probabilities, not probability densities, are used to describe their distributions. For example, firing rates obtained by counting spikes over the duration of a trial take discrete values and can be described by a probability. However, we sometimes treat the response firing rates or the stimulus values as continuous variables. In this case, the probabilities listed must be replaced by the corresponding probability densities, $p[\mathbf{r}]$, $p[\mathbf{r}|s]$, etc. Nevertheless, the relationships discussed above are equally valid.

the estimate v to its true value $v^{\mathbf{M}}$ under a set of conditions discussed in the texts mentioned in the annotated bibliography.

The other half of policy iteration is policy improvement. This normally works by finding an action a^* that maximizes the expression in the curly brackets in equation 9.33 and making the new $P_{\mathbf{M}}[a^*; u]=1$. One can show that the new policy will be uniformly better than the old policy, making the expected long-term reward at every state no smaller than the old policy, or equally large, if it is already optimal. Further, because the number of different policies is finite, policy iteration is bound to converge.

Performing policy improvement like this requires knowledge of the transition probabilities and mean rewards. Reinforcement learning again uses an asynchronous, model-free approach to policy improvement, using Monte Carlo samples. First, note that any policy $\mathbf{M}'$ that improves the average value

$$\sum_a P_{\mathbf{M}'}[u; a] \left\{ \langle r_a(u) \rangle + \sum_{u'} P[u'|u; a] v^{\mathbf{M}}(u') \right\} \tag{9.35}$$

for every state u is guaranteed to be a better policy. The idea for a single state u is to treat equation 9.35 rather like equation 9.15, except replacing the average immediate reward $\langle r_a \rangle$ there by an effective average immediate reward $\langle r_a(u) \rangle + \sum_{u'} P[u'|u; a] v^{\mathbf{M}}(u')$ to take long-term as well as current rewards into account. By the same reasoning as above, $r_a(u) + v(u')$ is used as an approximate Monte Carlo sample of the effective immediate reward, and $v(u)$ as the equivalent of the reinforcement comparison term $\bar{r}$. This leads directly to the actor learning rule of equation 9.25.

Note that there is an interaction between the stochasticity in the reinforcement learning versions of policy evaluation and policy improvement. This means that it is not known whether the two together are guaranteed to converge. One could perform temporal difference policy evaluation (which can be proven to converge) until convergence before attempting policy improvement, and this would be sure to work.

9.7 Annotated Bibliography

Dickinson (1980), **Mackintosh (1983)**, and **Shanks (1995)** review animal and human conditioning behavior, including alternatives to Rescorla & Wagner's (1972) rule. **Gallistel (1990)** and Gallistel & Gibbon (2000) discuss aspects of conditioning, in particular to do with timing, that we have omitted.

Our description of the temporal difference model of classical conditioning in this chapter is based on **Sutton (1988)** and **Sutton & Barto (1990)**. The treatment of static action choice comes from **Narendra & Thatachar (1989)** and **Williams (1992)**, and that of action choice with delayed rewards and

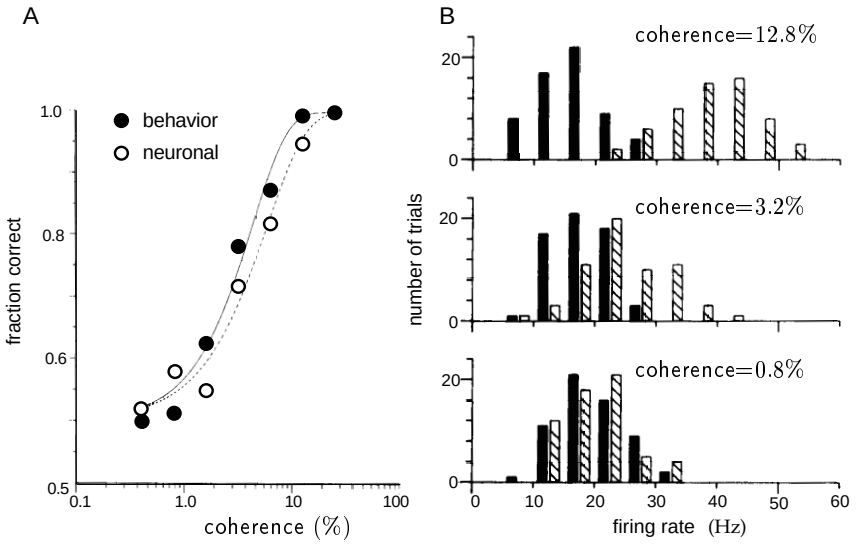

Figure 3.2 Behavioral and electrophysiological results from a random-dot motion-discrimination task. (A) The filled circles show the fraction of correct discriminations made by a monkey as a function of the degree of coherence of the motion. The open circles show the discrimination accuracy that an ideal observer could achieve on the analogous two-alternative forced-choice discrimination task, given the neural responses. (B) Firing-rate histograms for three different levels of coherence. Hatched rectangles show the results for motion in the plus direction, and solid rectangles, for motion in the minus direction. The histograms have been thinned for clarity so that not all the bins are shown. (Adapted from Britten et al., 1992.)

of motion in the random dot images. During the same task, recordings were made from neurons in area MT. Only two possible directions of coherent movement of the dots were used while a particular neuron was being recorded; either the direction that produced the maximum response in that neuron, or the opposite direction. The monkey's task was to discriminate between these two directions. The filled circles and solid curve in figure 3.2A show the proportion of correct responses in a typical experiment. Below 1% coherence, the responses were near chance (fraction correct = 0.5), but the monkey approached perfect performance (fraction correct = 1) above 10% coherence.

Figure 3.2B shows histograms of average firing rates in response to different levels of movement coherence. The firing rates plotted are the number of spikes recorded during the 2 s period that the stimulus was presented, divided by 2 s. The neuron shown tended to fire more spikes when the motion was in its preferred direction, which we will call the plus (or $+$) direction (hatched histogram), than in the other, minus (or $-$) direction (solid histogram). At high coherence levels, the firing-rate distributions corresponding to the two directions are fairly well separated, while at low coherence levels, they merge. Although spike count rates take only discrete values, it is more convenient to treat r as a continuous variable for our discussion. Treated as probability densities, these two distributions are

In the appendix, we show more precisely how temporal difference learning can be seen as a Monte Carlo technique for performing policy iteration.

9.6 Appendix

Markov Decision Problems

Markov decision problems offer a simple formalism for describing tasks such as the maze. A Markov decision problem is comprised of states, actions, transitions, and rewards. The states, labeled by u, are what we called locations in the maze task, and the actions, labeled by a, are the analogs of the choices of which direction to run. In the maze, each action taken at state u led uniquely and deterministically to a new state u'. Markov decision problems generalize this to include the possibility that the transitions from u due to action a may be stochastic, leading to state u' with a transition probability $P[u'|u;a]$. $\sum_{u'} P[u'|u;a] = 1$ for all u and a, because the animal has to end up somewhere. There can be absorbing states (like the shaded boxes in figure 9.7), which are u for which $P[u|u;a] = 1$ for all actions a (i.e., there is no escape for the animal from these locations). Finally, the rewards r can depend both on the state u and the action executed a, and they might be stochastic. We write $\langle r_a(u)\rangle$ for the mean reward in this case. The crucial Markov property is that, given the state at the current time step, the distribution over future states and rewards is independent of the past states. This defines the state sequence as the output of a controlled Markov chain. For convenience, we consider only Markov decision problems that are finite (finite numbers of actions and states) and absorbing (the animal always ends up in one of the absorbing states), and for which the rewards are bounded. We also require that $\langle r_a(u)\rangle = 0$ for all actions a at all absorbing states.

absorbing state

Markov property

The Bellman Equation

The task for a system or animal facing a Markov decision problem, starting in state u at time 0, is to choose a policy, denoted by $\mathbf{M}$, that maximizes the expected total future reward,

$$v^*(u) = \max_{\mathbf{M}} \left\langle \sum_{t=0}^{\infty} r_{a(t)}(u(t)) \right\rangle_{u,\mathbf{M}}, \tag{9.31}$$

where $u(0) = u$, actions $a(t)$ are determined (either deterministically or stochastically) on the basis of the state $u(t)$ according to policy $\mathbf{M}$, and the notation $\langle\rangle_{u,\mathbf{M}}$ implies taking an expectation over the actions and the states to which they lead, starting at state u and using policy $\mathbf{M}$.

The trouble with the sum in equation 9.31 is that the action $a(0)$ at time 0 affects not only $\langle r_{a(0)}(u(0))\rangle$, but, by influencing the state of the sys-

is near 0 and the power near 1. In general, it is impossible to choose the threshold so that both the size and the power of the test are optimized; a compromise must be made. A logical optimization criterion is to maximize the probability of getting a correct answer, which is equal to $(\beta(z) + 1 - \alpha(z))/2$ if the plus and minus stimuli occur with equal probability. Although this is a possible approach for the experiment we are studying, the analysis we present introduces a powerful technique that makes better use of the full range of recorded data and can be generalized to tasks where the optimal strategy is unknown. This approach makes use of ROC curves, which indicate how the size and power of a test trade off as the threshold is varied.

ROC Curves

receiver operating characteristic, ROC

The receiver operating characteristic (ROC) curve provides a way of evaluating how test performance depends on the choice of the threshold z. Each point on an ROC curve corresponds to a different value of z. The x coordinate of the point is α, the size of the test for this value of z, and the y coordinate is β, its power. As the threshold is varied continuously, these points trace out the ROC plot. If $z = 0$, the firing rate will always be greater than or equal to z, so the decoding procedure will always give the answer "plus". Thus, for $z = 0$, $\alpha = \beta = 1$, producing a point at the upper-right corner of the ROC plot. At the other extreme, if z is very large, r will always be less than z, the test will always report "minus", and $\alpha = \beta = 0$. This produces a point at the bottom-left corner of the plot. Between these extremes, a curve is traced out as a function of z.

Figure 3.3 shows ROC curves computed by Britten et al. for several different values of the stimulus coherence. At high coherence levels, when the task is easy, the ROC curve rises rapidly from $\alpha(z) = 0$, $\beta(z) = 0$ as the threshold is lowered from a high value, and the probability $\beta(z)$ of a correct "plus" answer quickly approaches 1 without a concomitant increase in $\alpha(z)$. As the threshold is lowered further, the probability of giving the answer "plus" when the correct answer is "minus" also rises, and $\alpha(z)$ increases. When the task is difficult, the curve rises more slowly as z is lowered; and if the task is impossible, in that the test merely gives random answers, the curve will lie along the diagonal $\alpha = \beta$, because the probabilities of answers being correct and incorrect are equal. This is exactly the trend of the ROC curves at different coherence levels shown in figure 3.3.

Examination of figure 3.3 suggests a relationship between the area under the ROC curve and the level of performance on the task. When the ROC curve in figure 3.3 lies along the diagonal, the area underneath it is $1/2$, which is the probability of a correct answer in this case (given any threshold). When the task is easy and the ROC curve hugs the left axis and upper limit in figure 3.3, the area under it approaches 1, which is again the probability of a correct answer (given an appropriate threshold). However, the precise relationship between task performance and the area under the

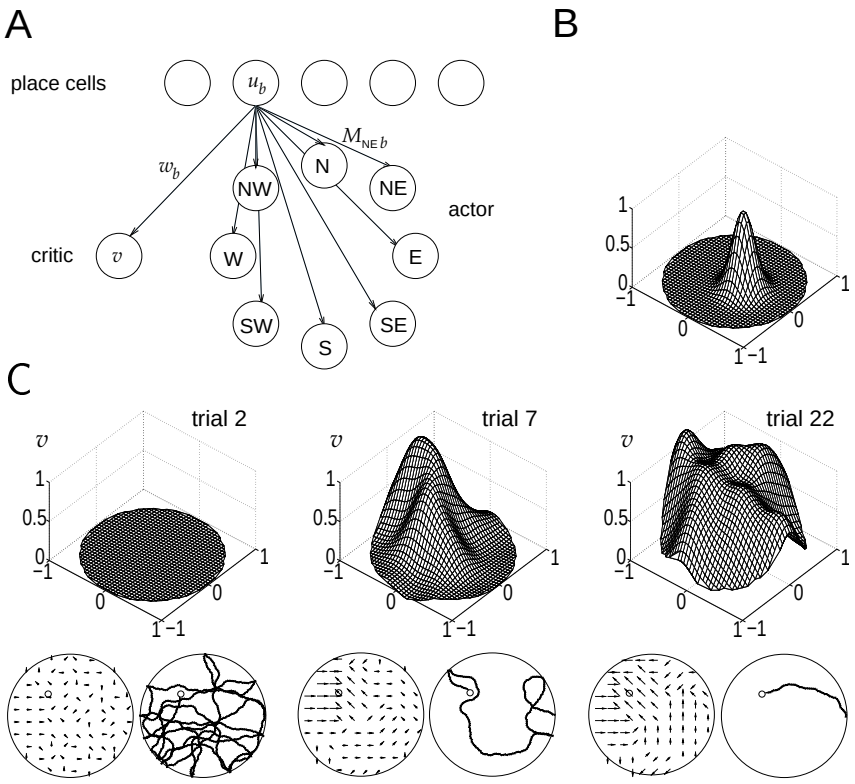

Figure 9.10 Reinforcement learning model of a rat solving a simple water maze task in a 2 m diameter circular pool. (A) There are 493 place cell inputs and 8 actions. The rat moves at 0.3 m/s and reflects off the walls of the maze if it hits them. (B) Gaussian place field for a single input cell with width 0.16 m. The centers of the place fields for different cells are uniformly distributed across the pool. (C) Upper: The development of the value function v as a function of the location in the pool over the first 22 trials, starting from $v = 0$ everywhere . Lower arrow plots: The action with the highest probability for each location in the maze. Lower path plots: Actual paths taken by the model rat from random starting points to the platform, indicated by a small circle. A slight modification of the actor learning rule was used to enforce generalization between spatially similar actions. (Adapted from Foster et al., 2000.)

the platform indicated by a small circle in the lower part of figure 9.10C. At that point a reward of 1 is provided. The reward is discounted with $\gamma = 0.9975$ to model the incentive for the rat to find the goal as quickly as possible. Figure 9.10C indicates the course of learning (trials 2, 7, and 22) of the expected future reward as a function of location (upper figures) and the policy (lower figures with arrows). The lower figures also show sample paths taken by the rat (lower figures with wiggly lines). The final value function (at trial 22) is rather inaccurate, but nevertheless the policy learned is broadly correct, and the paths to the platform are quite short and direct.

Judged by measures such as path length, initial learning proceeds in the

probability of being correct if the order of the stimuli is reversed.

The probability that $r \geq z$ for a minus stimulus, which is just $\alpha(z)$, can be written as an integral of the conditional firing-rate probability density $p[r|-]$,

$$\alpha(z) = \int_z^\infty dr\, p[r|-]\,. \tag{3.7}$$

Taking the derivative of this equation with respect to z, we find that

$$\frac{d\alpha}{dz} = -p[z|-]\,. \tag{3.8}$$

This allows us to make the replacement $dz\, p[z|-] \to -d\alpha$ in the integral of equation 3.6 and to change the integration variable from z to α. Noting that $\alpha = 1$ when $z = 0$ and $\alpha = 0$ when $z = \infty$, we find

$$P[\text{correct}] = \int_0^1 d\alpha\, \beta\,. \tag{3.9}$$

The ROC curve is just β plotted as a function of α, so this integral is the area under the ROC curve. Thus, the area under the ROC curve is the probability of responding correctly in the two-alternative forced-choice test.

Suppose that $p[r|+]$ and $p[r|-]$ are both Gaussian functions with means $\langle r\rangle_+$ and $\langle r\rangle_-$, and a common variance σ_r^2. The reader is invited to show that, in this case,

$$P[\text{correct}] = \frac{1}{2}\text{erfc}\left(\frac{\langle r\rangle_- - \langle r\rangle_+}{2\sigma_r}\right) = \frac{1}{2}\text{erfc}\left(-\frac{d'}{2}\right)\,, \tag{3.10}$$

complementary error function

where d' is the discriminability defined in equation 3.4 and $\text{erfc}(x)$ is the complementary error function (which is an integral of a Gaussian distribution) defined as

$$\text{erfc}(x) = \frac{2}{\sqrt{\pi}}\int_x^\infty dy\, \exp(-y^2)\,. \tag{3.11}$$

In the case where the distributions are equal-variance Gaussians, the relationship between the discriminability and the area under the ROC curve is invertible because the complementary error function is monotonic. It is common to quote d' values even for non-Gaussian distributions by inverting the relationship between $P[\text{correct}]$ and d' in equation 3.10.

ROC Analysis of Motion Discrimination

To interpret their experiment as a two-alternative forced-choice task, Britten et al. imagined that, in addition to being given the firing rate of the recorded neuron during stimulus presentation, the observer is given the firing rate of a hypothetical "anti-neuron" having response characteristics

u, can depend on it. The simplest dependence is provided by the linear form $v(u) = \mathbf{w} \cdot \mathbf{u}(u)$, similar to the input-output relationship used in linear feedforward network models. The learning rule for the critic (equation 9.24) is then generalized to include the information provided by the state vector,

$$\mathbf{w} \rightarrow \mathbf{w} + \epsilon \delta \mathbf{u}(u) \,, \tag{9.26}$$

with δ given as in equation 9.24. The maze task we discussed could be formulated in this way by using what is called a unary representation, $\mathbf{u}(\mathrm{A}) = (1, 0, 0)$, $\mathbf{u}(\mathrm{B}) = (0, 1, 0)$, and $\mathbf{u}(\mathrm{C}) = (0, 0, 1)$.

unary representation

We must also modify the actor learning rule to make use of the information provided by the state vector. This is done by generalizing the action value vector $\mathbf{m}$ to a matrix $\mathbf{M}$, called an action matrix. $\mathbf{M}$ has as many columns as there are components of $\mathbf{u}$ and as many rows as there are actions. Given input $\mathbf{u}$, action a is chosen at location u with the softmax probability of equation 9.12, but using component a of the action value vector,

action matrix $\mathbf{M}$

$$\mathbf{m} = \mathbf{M} \cdot \mathbf{u}(u) \quad \text{or} \quad m_a = \sum_b M_{ab} u_b(u) \,. \tag{9.27}$$

In this case, the learning rule 9.25 must be generalized to specify how to change elements of the action matrix when action a is chosen at location u with state vector $\mathbf{u}(u)$, leading to location u'. A rule similar to equation 9.25 is appropriate, except that the change in $\mathbf{M}$ depends on the state vector $\mathbf{u}$,

$$M_{a'b} \rightarrow M_{a'b} + \epsilon \left(\delta_{aa'} - P[a'; u] \right) \delta u_b(u) \tag{9.28}$$

three-term covariance rule

for all a', with δ given again as in equation 9.24. This is called a three-term covariance learning rule.

We can speculate about the biophysical significance of the three-term covariance rule by interpreting $\delta_{aa'}$ as the output of cell a' when action a is chosen (which has mean value $P[a'; u]$) and interpreting $\mathbf{u}$ as the input to that cell. Compared with the Hebbian covariance rules studied in chapter 8, learning is gated by a third term, the reinforcement signal δ. It has been suggested that the dorsal striatum, which is part of the basal ganglia, is involved in the selection and sequencing of actions. Terminals of axons projecting from the substantia nigra pars compacta release dopamine onto synapses within the striatum, suggesting that they might play such a gating role. The activity of these dopamine neurons is similar to that of the VTA neurons discussed previously as a possible substrate for δ.

dorsal striatum
basal ganglia

The second generalization is to the case that rewards and punishments received soon after an action are more important than rewards and punishments received later. One natural way to accommodate this is a technique called exponential discounting. In computing the expected future reward, this amounts to multiplying a reward that will be received τ time steps after a given action by a factor γ^τ, where $0 \leq \gamma \leq 1$ is the discounting factor. The smaller γ, the stronger the effect (i.e., the less important temporally

discounting

Combining this result with 3.8, we find that

$$\frac{d\beta}{d\alpha} = \frac{d\beta}{dz}\frac{dz}{d\alpha} = \frac{p[z|+]}{p[z|-]} = l(z)\,, \tag{3.14}$$

so the slope of the ROC curve is equal to the likelihood ratio.

Another way of seeing that comparing the likelihood ratio to a threshold value is an optimal decoding procedure for discrimination uses a Bayesian approach based on associating a cost or penalty with getting the wrong answer. Suppose that the penalty associated with answering "minus" when the correct answer is "plus" is quantified by the loss parameter L_-. Similarly, quantify the loss for answering "plus" when the correct answer is "minus" as L_+. For convenience, we assume that there is neither loss nor gain for answering correctly. The probabilities that the correct answer is "plus" or "minus", given the firing rate r, are $P[+|r]$ and $P[-|r]$ respectively. These probabilities are related to the conditional firing-rate probability densities by Bayes theorem,

$$P[+|r] = \frac{p[r|+]P[+]}{p[r]} \quad \text{and} \quad P[-|r] = \frac{p[r|-]P[-]}{p[r]}\,. \tag{3.15}$$

The average loss expected for a "plus" answer when the firing rate is r is the loss associated with being wrong times the probability of being wrong, $\text{Loss}_+ = L_+P[-|r]$. Similarly, the expected loss when answering "minus" is $\text{Loss}_- = L_-P[+|r]$. A reasonable strategy is to cut the losses, answering "plus" if $\text{Loss}_+ \leq \text{Loss}_-$ and "minus" otherwise. Using equation 3.15, we find that this strategy gives the response "plus" if

$$l(r) = \frac{p[r|+]}{p[r|-]} \geq \frac{L_+}{L_-}\frac{P[-]}{P[+]}\,. \tag{3.16}$$

This shows that the strategy of comparing the likelihood ratio to a threshold is a way of minimizing the expected loss. The right side of this inequality gives an explicit formula for the value of the threshold that should be used, and reflects two factors. One is the relative losses for the two sorts of possible errors. The other is the prior probabilities that the stimulus is plus or minus. Interestingly, it is possible to change the thresholds that human subjects use in discrimination tasks by manipulating these two factors.

If the conditional probability densities $p[r|+]$ and $p[r|-]$ are Gaussians with means r_+ and r_- and identical variances σ_r^2, and $P[+] = P[-] = 1/2$, the probability $P[+|r]$ is a sigmoidal function of r,

$$P[+|r] = \frac{1}{1+\exp(-d'(r-r_{\text{ave}})/\sigma_r)}\,, \tag{3.17}$$

where $r_{\text{ave}} = (r_+ + r_-)/2$. This provides an alternate interpretation of the parameter d' that is often used in the psychophysics literature; it determines the slope of a sigmoidal function fitted to $P[+|r]$.

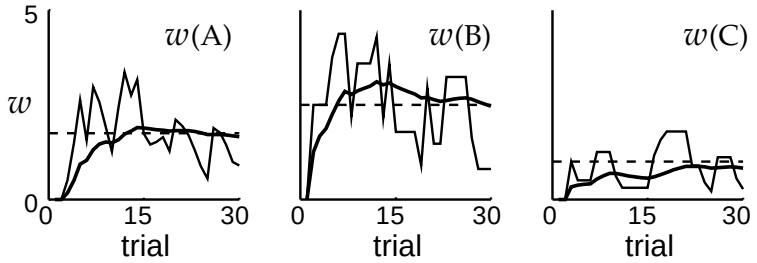

Figure 9.8 Policy evaluation. The thin lines show the course of learning of the weights $w(\mathrm{A})$, $w(\mathrm{B})$, and $w(\mathrm{C})$ over trials through the maze in figure 9.7, using a random unbiased policy ($\mathbf{m}(u) = 0$). Here $\epsilon = 0.5$, so learning is fast but noisy. The dashed lines show the correct weight values from equation 9.23. The thick lines are running averages of the weight values.

Policy Improvement

In policy improvement, the expected total future rewards at the different locations are used as surrogate immediate rewards. Suppose the rat is about to take action a at location u and move to location u'. The expected worth to the rat of that action is the sum of the actual reward received and the rewards that are expected to follow, which is $r_a(u) + v(u')$. For simplicity, we assume that the rat receives the reward for location u at the same time it decides to move on to location u'. The direct actor scheme of equation 9.22 uses the difference $r_a - \bar{r}$ between a sample of the worth of the action (r_a) and a reinforcement comparison term ($\bar{r}$), which might be the average value over all the actions that can be taken. Policy improvement uses $r_a(u) + v(u')$ as the equivalent of the sampled worth of the action, and $v(u)$ as the average value across all actions that can be taken at u. The difference between these is $\delta = r_a(u) + v(u') - v(u)$, which is exactly the same term as in policy evaluation (equation 9.24). The policy improvement or actor learning rule is then *actor learning rule*

$$m_{a'}(u) \rightarrow m_{a'}(u) + \epsilon \left(\delta_{aa'} - P[a'; u]\right) \delta \tag{9.25}$$

for all a', where $P[a'; u]$ is the probability of taking action a' at location u given by the softmax distribution of equation 9.12 with action value $m_{a'}(u)$.

To look at this more concretely, consider the temporal difference error starting from location $u = \mathrm{A}$, using the true values of the locations given by equation 9.23 (i.e., assuming that policy evaluation is perfect). Depending on the action, δ takes the two values

$$\begin{aligned} \delta &= 0 + v(\mathrm{B}) - v(\mathrm{A}) = \quad 0.75 \quad \text{for a left turn} \\ \delta &= 0 + v(\mathrm{C}) - v(\mathrm{A}) = -0.75 \quad \text{for a right turn.} \end{aligned}$$

The learning rule of equation 9.25 increases the probability that the action with $\delta > 0$ is taken and decreases the probability that the action with $\delta < 0$

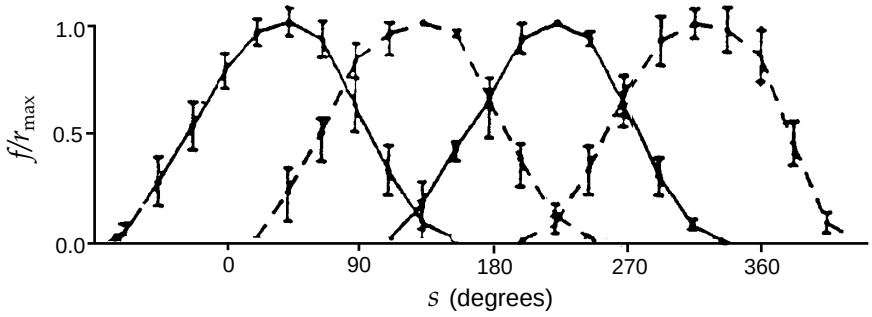

Figure 3.4 Tuning curves for the four low-velocity interneurons of the cricket cercal system plotted as a function of the wind direction s. Each neuron responds with a firing rate that is closely approximated by a half-wave rectified cosine function. The preferred directions of the neurons are located 90° from each other, and r_{max} values are typically around 40 Hz. Error bars show standard deviations. (Adapted from Theunissen and Miller, 1991.)

from their hind ends. These are covered with hairs that are deflected by air currents. Each hair is attached to a neuron that fires when the hair is deflected. Thousands of these primary sensory neurons send axons to a set of interneurons that relay the sensory information to the rest of the cricket's nervous system. No single interneuron of the cercal system responds to all wind directions, and multiple interneurons respond to any given wind direction. This implies that the interneurons encode the wind direction collectively as a population.

Theunissen and Miller (1991) measured both the mean and the variance of responses of cercal interneurons while blowing air currents at the cerci. At low wind velocities, information about wind direction is encoded by just four interneurons. Figure 3.4 shows average firing-rate tuning curves for the four relevant interneurons as a function of wind direction. These neurons are sensitive primarily to the angle of the wind around the vertical axis and not to its elevation above the horizontal plane. Wind speed was held constant in these experiments, so we do not discuss how it is encoded. The interneuron tuning curves are well approximated by half-wave rectified cosine functions. Neuron a (where $a = 1, 2, 3, 4$) responds with a maximum average firing rate when the angle of the wind direction is s_a, the preferred-direction angle for that neuron. The tuning curve for interneuron a in response to wind direction s, $\langle r_a \rangle = f_a(s)$, normalized to its maximum, can be written as

cosine tuning

$$\left(\frac{f(s)}{r_{max}}\right)_a = [(\cos(s - s_a)]_+ , \tag{3.20}$$

where the half-wave rectification eliminates negative firing rates. Here r_{max}, which may be different for each neuron, is a constant equal to the maximum average firing rate. The fit can be improved somewhat by introducing a small offset rate, but the simple cosine is adequate for our purposes.

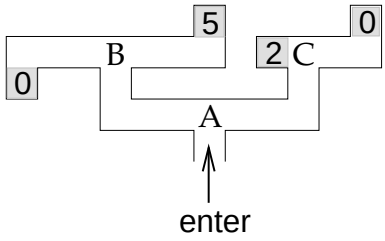

Figure 9.7 The maze task. The rat enters the maze from the bottom and has to move forward. Upon reaching one of the end points (the shaded boxes), it receives the number of food pellets indicated and the trial ends. Decision points are A, B, and C.

until after the rat also goes right at B.

There is an extensive body of theory in engineering, called dynamic programming, as to how systems of any sort can come to select appropriate actions in optimizing control problems similar to (and substantially more complicated than) the maze task. An important method on which we focus is called policy iteration. Our reinforcement learning version of policy iteration maintains and improves a stochastic policy, which determines the actions at each decision point (i.e., left or right turns at A, B, or C) through action values and the softmax distribution of equation 9.12. Policy iteration involves two elements. One, called the critic, uses temporal difference learning to estimate the total future reward that is expected when starting from A, B, or C, when the current policy is followed. The other element, called the actor, maintains and improves the policy. Adjustment of the action values at point A is based on predictions of the expected future rewards associated with points B and C that are provided by the critic. In effect, the rat learns the appropriate action at A, using the same methods of static action choice that allow it to learn the appropriate actions at B and C. However, rather than using an immediate reward as the reinforcement signal, it uses the expectations about future reward that are provided by the critic.

dynamic programming

policy iteration

critic

actor

The Maze Task

As we mentioned when discussing the direct actor, a stochastic policy is a way of assigning a probability distribution over actions (in this case choosing to turn either left or right) to each location (A, B, or C). The location is specified by a variable u that takes the values A, B, or C, and a two-component action value vector $\mathbf{m}(u)$ is associated with each location. The components of the action vector $\mathbf{m}(u)$ control the probability of taking a left or a right turn at u through the softmax distribution of equation 9.12.

The immediate reward provided when action a is taken at location u is written as $r_a(u)$. For the maze of figure 9.7, this takes values 0, 2, or 5, depending on the values of u and a. The predicted future reward expected

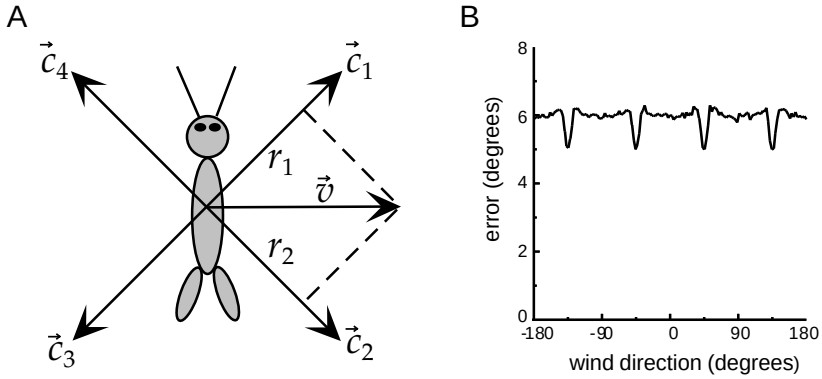

Figure 3.5 (A) Preferred directions of four cercal interneurons in relation to the cricket's body. The firing rate of each neuron for a fixed wind speed is proportional to the projection of the wind velocity vector $\vec{v}$ onto the preferred-direction axis of the neuron. The projection directions $\vec{c}_1$, $\vec{c}_2$, $\vec{c}_3$, and $\vec{c}_4$ for the four neurons are separated by 90°, and they collectively form a Cartesian coordinate system. (B) The root-mean-square error in the wind direction determined by vector decoding of the firing rates of four cercal interneurons. These results were obtained through simulation by randomly generating interneuron responses to a variety of wind directions, with the average values and trial-to-trial variability of the firing rates matched to the experimental data. The generated rates were then decoded using equation 3.22 and compared to the wind direction used to generate them. (B adapted from Salinas and Abbott, 1994.)

fact that one of the neurons responds maximally; rather, they arise because the two neurons with tuning curves adjacent to the maximally responding neuron are most sensitive to wind direction at these points.

As discussed in chapter 1, tuning curves of certain neurons in the primary motor cortex (M1) of the monkey can be described by cosine functions of arm movement direction. Thus, a vector decomposition similar to that of the cercal system appears to take place in M1. Many M1 neurons have nonzero offset rates, r_0, so they can represent the cosine function over most or all of its range. When an arm movement is made in the direction represented by a vector of unit length, $\vec{v}$, the average firing rates for such an M1 neuron, labeled by an index a (assuming that it fires over the entire range of angles), can be written as

$$\left(\frac{\langle r\rangle - r_0}{r_{\max}}\right)_a = \left(\frac{f(s) - r_0}{r_{\max}}\right)_a = \vec{v}\cdot\vec{c}_a\,, \tag{3.23}$$

where $\vec{c}_a$ is the preferred-direction vector that defines the selectivity of the neuron. Because these firing rates represent the full cosine function, it would, in principle, be possible to encode all movement directions in three dimensions using just three neurons. Instead, many thousands of M1 neurons have arm-movement-related tuning curves, resulting in a highly redundant representation. Of course, these neurons encode additional movement-related quantities; for example, their firing rates depend on the initial position of the arm relative to the body as well as on movement ve-

A similar expression applies to $\partial\langle r\rangle/\partial m_{\mathrm{y}}$, except that the blue and yellow labels are interchanged.

In stochastic gradient ascent, the changes in the parameter m_{b} are determined such that, averaged over trials, they end up proportional to $\partial\langle r\rangle/\partial m_{\mathrm{b}}$. We can derive a stochastic gradient ascent rule for m_{b} from equation 9.19 in two steps. First, we interpret the two terms on the right side as changes associated with the choice of blue and yellow flowers. This accounts for the factors $P[\mathrm{b}]$ and $P[\mathrm{y}]$, respectively. Second, we note that over trials in which blue is selected, $r_{\mathrm{b}} - \overline{r}$ averages to $\langle r_{\mathrm{b}}\rangle - \overline{r}$, and over trials in which yellow is selected, $r_{\mathrm{y}} - \overline{r}$ averages to $\langle r_{\mathrm{y}}\rangle - \overline{r}$. Thus, if we change m_{b} according to

$$\begin{aligned} m_{\mathrm{b}} &\to m_{\mathrm{b}} + \epsilon(1 - P[\mathrm{b}])(r_{\mathrm{b}} - \overline{r}) \quad \text{if b is selected} \\ m_{\mathrm{b}} &\to m_{\mathrm{b}} - \epsilon P[\mathrm{b}]\ (r_{\mathrm{y}} - \overline{r}) \qquad \text{if y is selected,} \end{aligned}$$

the average change in m_{b} is proportional to $\partial\langle r\rangle/\partial m_{\mathrm{b}}$. Note that m_b is changed even when the bee chooses the yellow flower. We can summarize this learning rule as

$$m_{\mathrm{b}} \to m_{\mathrm{b}} + \epsilon(\delta_{a\mathrm{b}} - P[\mathrm{b}])(r_a - \overline{r})\,, \tag{9.20}$$

where a is the action selected (either b or y) and $\delta_{a\mathrm{b}}$ is the Kronecker delta, $\delta_{a\mathrm{b}} = 1$ if $a = \mathrm{b}$ and $\delta_{a\mathrm{b}} = 0$ if $a = \mathrm{y}$. Similarly, the rule for m_{y} is

$$m_{\mathrm{y}} \to m_{\mathrm{y}} + \epsilon(\delta_{a\mathrm{y}} - P[\mathrm{y}])(r_a - \overline{r})\,. \tag{9.21}$$

The learning rule of equations 9.20 and 9.21 performs stochastic gradient ascent on the average reward, whatever the value of $\bar{r}$. Different values of $\bar{r}$ lead to different variances of the stochastic gradient terms, and thus different speeds of learning. A reasonable value for $\bar{r}$ is the mean reward under the specified policy or some estimate of this quantity.

Figure 9.6 shows the consequences of using the direct actor in the same stochastic foraging task as in figure 9.4. Two sample sessions are shown with widely differing levels of performance. Compared to the indirect actor, initial learning is quite slow, and the behavior after the reward characteristics of the flowers are interchanged can be poor. Explicit control of the trade-off between exploration and exploitation is difficult, because the action values can scale up to compensate for different values of β. Despite its comparatively poor performance in this task, the direct actor will be useful later as a model for how action choice can be separated from action evaluation.

The direct actor learning rule can be extended to multiple actions, $a = 1, 2, \ldots, N_{\mathrm{a}}$, by using the multidimensional form of the softmax distribution (equation 9.12). In this case, when action a is taken, $m_{a'}$ for all values of a' is updated according to

$$m_{a'} \to m_{a'} + \epsilon\left(\delta_{aa'} - P[a']\right)(r_a - \bar{r})\,. \tag{9.22}$$

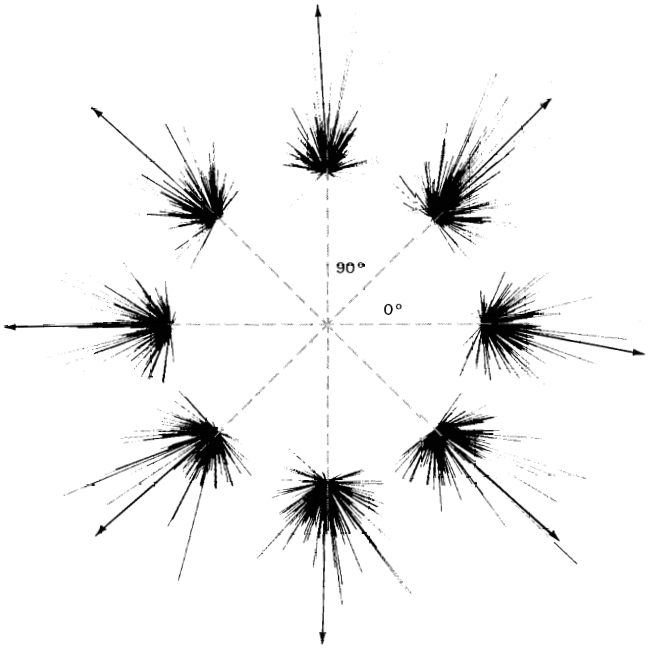

Figure 3.6 Comparison of population vectors with actual arm movement directions. Results are shown for eight different movement directions. Actual arm movement directions are radially outward at angles that are multiples of 45°. The groups of lines without arrows show the preferred-direction vectors of the recorded neurons multiplied by their firing rates. Vector sums of these terms for each movement direction are indicated by the arrows. The fact that the arrows point approximately radially outward shows that the population vector reconstructs the actual movement direction fairly accurately. (Figure adapted from Kandel et al., 1991, based on data from Kalaska et al., 1983.)

needed for a continuous stimulus parameter, $p[s|\mathbf{r}]$, can be obtained from the encoding probability density $p[\mathbf{r}|s]$ by the continuous version of Bayes theorem (equation 3.3),

$$p[s|\mathbf{r}] = \frac{p[\mathbf{r}|s]p[s]}{p[\mathbf{r}]}\,. \tag{3.26}$$

A disadvantage of these methods is that extracting $p[s|\mathbf{r}]$ from experimental data can be difficult. In contrast, the vector method only requires us to know the preferred stimulus values of the encoding neurons.

Bayesian inference

As mentioned in the previous paragraph, Bayesian inference is based on the minimization of a particular loss function $L(s, s_{\text{bayes}})$ that quantifies the "cost" of reporting the estimate s_{bayes} when the correct answer is s. The loss function provides a way of defining the optimality criterion for decoding analogous to the loss computation discussed previously for optimal discrimination. The value of s_{bayes} is chosen to minimize the expected loss averaged over all stimuli for a given set of rates, that is, to minimize the function $\int ds\, L(s, s_{\text{bayes}})p[s|\mathbf{r}]$. If the loss function is the squared differ-

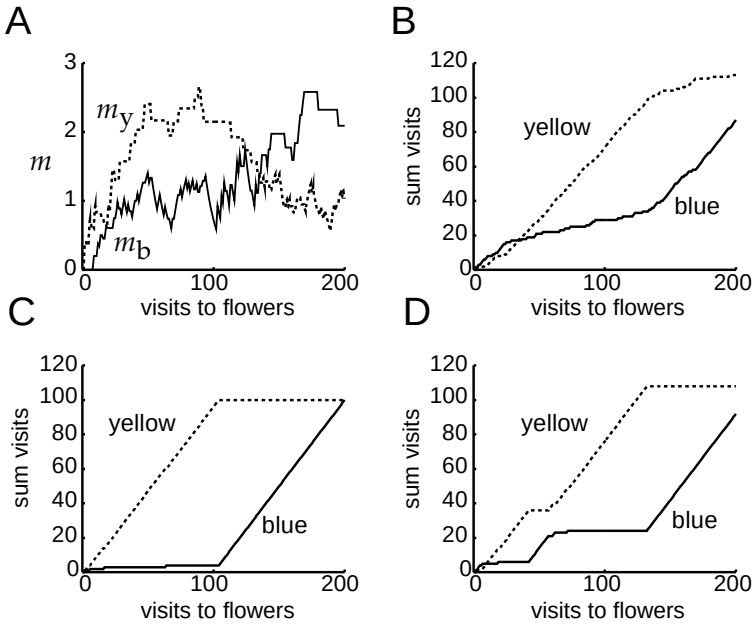

Figure 9.4 The indirect actor. Rewards were $\langle r_b \rangle = 1$, $\langle r_y \rangle = 2$ for the first 100 flower visits, and $\langle r_b \rangle = 2$, $\langle r_y \rangle = 1$ for the second 100 flower visits. Nectar was delivered stochastically on half the flowers of each type. (A) Values of m_b (solid) and m_y (dashed) as a function of visits for $\beta = 1$. Because a fixed value of $\epsilon = 0.1$ was used, the weights do not converge perfectly to the corresponding average reward, but they fluctuate around these values. (B-D) Cumulative visits to blue (solid) and yellow (dashed) flowers. (B) When $\beta = 1$, learning is slow, but ultimately the change to the optimal flower color is made reliably. (C, D) When $\beta = 50$, sometimes the bee performs well (C), and other times it performs poorly (D).

reward is the same for the two flower types). Between trials 15 and 16, the delivery characteristics of the flowers were swapped. Figure 9.5A shows the average performance of five bees on this task in terms of their percentage visits to the blue flowers across trials. They exhibit a strong preference for the constant flower type and switch this preference within only a few visits to the flowers when the contingencies change.

To apply the foraging model we have been discussing to the experiment shown in figure 9.5A, we need to model the risk avoidance exhibited by the bees, that is, their reluctance to choose the unreliable flower. One way to do this is to assume that the bees base their policy on the subjective utility function of the nectar volume shown in figure 9.5B, rather than on the nectar volume itself. Because the function is concave, the mean utility of the unreliable flowers is less than that of the reliable flowers. Figure 9.5C shows that the choices of the model bee match those of the real bees quite well. The model bee is less variable than the actual bees (even more than it appears, because the curve in 9.5A is averaged over five bees), perhaps because the model bees are not sampling from a two-dimensional array of flowers.

subjective utility

The Bayesian result has a slightly smaller average error across all angles. The dips in the error curves in figure 3.7, as in the curve of figure 3.5B, appear at angles where one tuning curve peaks and two others rise from threshold (see figure 3.4). As in figure 3.5B, these dips are due to the two neurons responding near threshold, not to the maximally responding neuron. They occur because neurons are most sensitive at points where their tuning curves have maximum slopes, which in this case is near threshold (see figure 3.11).

Comparing these results with figure 3.5B shows the improved performance of these methods relative to the vector method. The vector method performs extremely well for this system, so the degree of improvement is not large. This is because the cercal responses are well described by cosine functions and their preferred directions are 90° apart. Much more dramatic differences occur when the tuning curves are not cosines or the preferred stimulus directions are not perpendicular.

Up to now, we have considered the decoding of a direction angle. We now turn to the more general case of decoding an arbitrary continuous stimulus parameter. An instructive example is provided by an array of N neurons with preferred stimulus values distributed uniformly across the full range of possible stimulus values. An example of such an array for Gaussian tuning curves,

$$f_a(s) = r_{\max} \exp\left(-\frac{1}{2}\left(\frac{s - s_a}{\sigma_a}\right)^2\right), \tag{3.28}$$

is shown in figure 3.8. In this example, each neuron has a tuning curve with a different preferred value s_a and potentially a different width σ_a (although all the curves in figure 3.8 have the same width). If the tuning curves are evenly and densely distributed across the range of s values, the sum of all tuning curves $\sum f_a(s)$ is approximately independent of s. The roughly flat line in figure 3.8 is proportional to this sum. The constancy of the sum over tuning curves will be useful in the following analysis.

Tuning curves give the mean firing rates of the neurons across multiple trials. In any single trial, measured firing rates will vary from their mean values. To implement the Bayesian, MAP, or ML approach, we need to know the conditional firing-rate probability density $p[\mathbf{r}|s]$ that describes this variability. We assume that the firing rate r_a of neuron a is determined by counting n_a spikes over a trial of duration T (so that $r_a = n_a/T$), and that the variability can be described by the homogeneous Poisson model discussed in chapter 1. In this case, the probability of stimulus s evoking $n_a = r_a T$ spikes, when the average firing rate is $\langle r_a \rangle = f_a(s)$, is given by (see chapter 1)

$$P[r_a|s] = \frac{(f_a(s)T)^{r_a T}}{(r_a T)!} \exp(-f_a(s)T)\,. \tag{3.29}$$

If we assume that each neuron fires independently, the firing-rate proba-

We treat a simplified version of the problem, ignoring the spatial aspects of sampling, and assuming that a model bee is faced with repeated choices between two different flowers. If the bee chooses the blue flower on a trial, it receives a quantity of nectar r_b drawn from a probability density $p[r_\mathrm{b}]$. If it chooses the yellow flower, it receives a quantity r_y, drawn from a probability density $p[r_\mathrm{y}]$. The task of choosing between the flowers is a form of stochastic two-armed bandit problem (named after slot machines), and is formally equivalent to many instrumental conditioning tasks.

two-armed bandit

The model bee has a stochastic policy, which means that it chooses blue and yellow flowers with probabilities that we write as $P[\mathrm{b}]$ and $P[\mathrm{y}]$, respectively. A convenient way to parameterize these probabilities is to use the softmax distribution

stochastic policy

softmax

$$P[\mathrm{b}] = \frac{\exp(\beta m_\mathrm{b})}{\exp(\beta m_\mathrm{b}) + \exp(\beta m_\mathrm{y})} \quad P[\mathrm{y}] = \frac{\exp(\beta m_\mathrm{y})}{\exp(\beta m_\mathrm{b}) + \exp(\beta m_\mathrm{y})} . \tag{9.11}$$

Here, m_b and m_y are parameters, known as action values, that are adjusted by one of the learning processes described below. Note that $P[\mathrm{b}] + P[\mathrm{y}] = 1$, corresponding to the fact that the model bee invariably makes one of the two choices. Also note that $P[\mathrm{b}] = \sigma(\beta(m_\mathrm{b} - m_\mathrm{y}))$, where $\sigma(m) = 1/(1 + \exp(-m))$ is the standard sigmoid function, which grows monotonically from 0 to 1 as m varies from $-\infty$ to ∞. $P[\mathrm{y}]$ is similarly a sigmoid function of $\beta(m_\mathrm{y} - m_\mathrm{b})$. The parameters m_b and m_y determine the frequency at which blue and yellow flowers are visited. Their values must be adjusted during the learning process on the basis of the reward provided.

action values m

The parameter β determines the variability of the bee's actions and exerts a strong influence on exploration. For large β, the probability of an action rises rapidly to 1, or falls rapidly to 0, as the difference between the action values increases or decreases. This makes the bee's action choice almost a deterministic function of the m variables. If β is small, the softmax probability approaches 1 or 0 more slowly, and the bee's actions are more variable and random. Thus, β controls the balance between exploration (small β) and exploitation (large β). The choice of whether to explore to determine if the current policy can be improved, or to exploit the available resources on the basis of the current policy, is known as the exploration-exploitation dilemma. Exploration is clearly critical, because the bee must sample from the two colors of flowers to determine which is better, and keep sampling to make sure that the reward conditions have not changed. But exploration is costly, because the bee has to sample flowers it believes to be less beneficial, to check if this is really the case. Some algorithms adjust β over trials, but we do not consider this possibility.

exploration-exploitation dilemma

There are only two possible actions in the example we study, but the extension to multiple actions, $a = 1, 2, \ldots, N_\mathrm{a}$, is straightforward. In this case, a vector $\mathbf{m}$ of parameters controls the decision process, and the probability $P[a]$ of choosing action a is

action value vector $\mathbf{m}$

$$P[a] = \frac{\exp(\beta m_a)}{\sum_{a'=1}^{N_\mathrm{a}} \exp(\beta m_{a'})} . \tag{9.12}$$

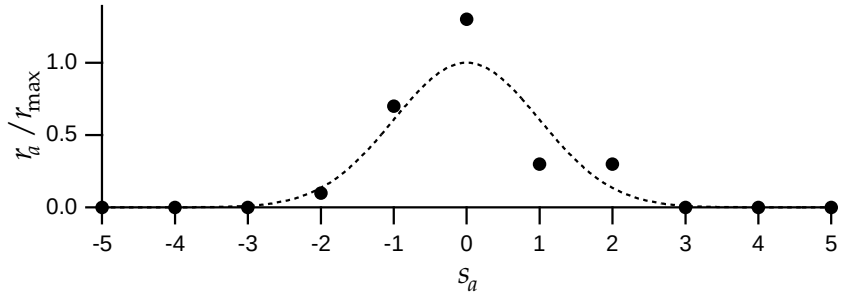

Figure 3.9 Simulated responses of 11 neurons with the Gaussian tuning curves shown in figure 3.8 to a stimulus value of 0. Firing rates for a single trial, generated using the Poisson model, are plotted as a function of the preferred-stimulus values of the different neurons in the population (filled circles). The dashed curve shows the tuning curve for the neuron with $s_a = 0$. Its heights at integer values of s_a are the average responses of the corresponding cells. It is possible to have $r_a > r_{max}$ (point at $s_a = 0$) because r_{max} is the maximum average firing rate, not the maximum firing rate.

mined by

$$\sum_{a=1}^{N} r_a \frac{f_a'(s_{ML})}{f_a(s_{ML})} = 0\,, \tag{3.32}$$

where the prime denotes a derivative. If the tuning curves are the Gaussians of equation 3.28, this equation can be solved explicitly using the result $f_a'(s)/f_a(s) = (s_a - s)/\sigma_a^2$,

$$s_{ML} = \frac{\sum r_a s_a/\sigma_a^2}{\sum r_a/\sigma_a^2}\,. \tag{3.33}$$

If all the tuning curves have the same width, this reduces to

$$s_{ML} = \frac{\sum r_a s_a}{\sum r_a}\,, \tag{3.34}$$

which is a simple estimation formula with an intuitive interpretation as the firing-rate weighted average of the preferred values of the encoding neurons. The numerator of this expression is reminiscent of the population vector.

Although equation 3.33 gives the ML estimate for a population of neurons with Poisson variability, it has some undesirable properties as a decoding algorithm. Consider a neuron with a preferred stimulus value s_a that is much greater than the actual stimulus value s. Because $s_a \gg s$, the average firing rate of this neuron is essentially 0. For a Poisson distribution, zero rate implies zero variability. If, however, this neuron fires one or more spikes on a trial due to a non-Poisson source of variability, this will cause a large error in the estimate because of the large weighting factor s_a.

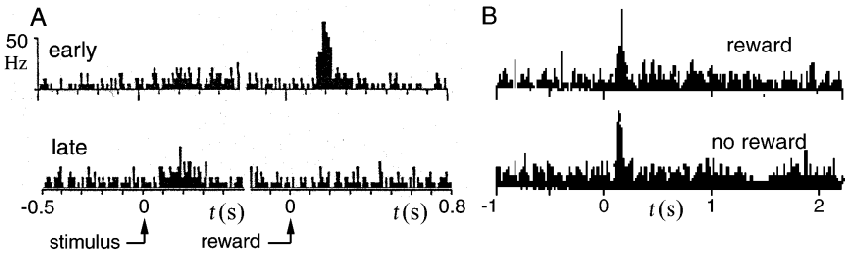

Figure 9.3 Activity of dopaminergic neurons in the VTA for a monkey performing reaction time tasks. (A) Activity of a dopamine cell accumulated over 20 trials showing the spikes time-locked to a stimulus (left panels) or to the reward (right panels) at the times marked 0. The top row is for early trials before the behavior is fully established. The bottom row is for late trials, when the monkey expects the reward on the basis of the stimulus. (B) Activity of a dopamine neuron with and without an expected reward delivery in a similar task. The top row shows the normal behavior of the cell when the reward is delivered. The bottom row shows the result of not delivering an expected reward. The basal firing rate of dopamine cells is rather low, but the inhibition at the time the reward would have been given is evident. (A adapted from Mirenowicz & Schultz, 1994; B adapted from Schultz, 1998.)

Dopamine and Predictions of Reward

The prediction error δ plays an essential role in both the Rescorla-Wagner and temporal difference learning rules, and we might hope to find a neural signal that represents this quantity. One suggestion is that the activity of dopaminergic neurons in the ventral tegmental area (VTA) in the midbrain plays this role.

ventral tegmental area VTA

There is substantial evidence that dopamine is involved in reward learning. Drugs of addiction, such as cocaine and amphetamines, act partly by increasing the longevity of the dopamine that is released onto target structures such as the nucleus accumbens. Other drugs, such as morphine and heroin, also affect the dopamine system. Further, dopamine delivery is important in self-stimulation experiments. Rats will compulsively press levers that cause current to be delivered through electrodes into various areas of their brains. One of the most effective self-stimulation sites is the medial forebrain ascending bundle, which is an axonal pathway. Stimulating this pathway is likely to cause increased delivery of dopamine to the nucleus accumbens and other areas of the brain, because the bundle contains many fibers from dopaminergic cells in the VTA projecting to the nucleus accumbens.

dopamine

In a series of studies by Schultz and his colleagues (Schultz, 1998), monkeys were trained through instrumental conditioning to respond to stimuli such as lights and sounds in order to obtain food and drink rewards. The activities of cells in the VTA were recorded while the monkeys learned these tasks. Figure 9.3A shows the activity of a dopamine cell at two times during learning. The figure is based on a reaction time task in which the monkey keeps a finger resting on a key until a sound is presented. The

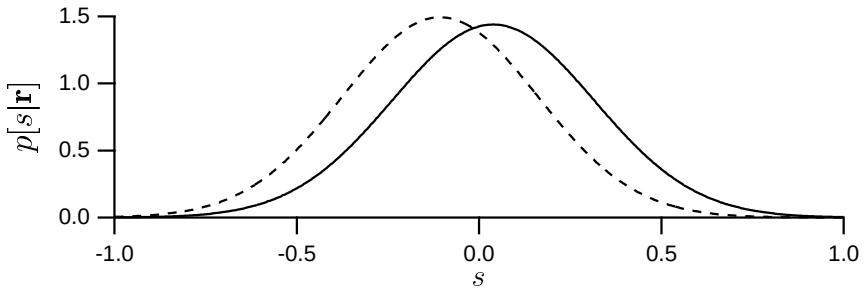

Figure 3.10 Probability densities for the stimulus, given the firing rates shown in figure 3.9 and assuming the tuning curves of figure 3.8. The solid curve is $p[s|\mathbf{r}]$ when the prior distribution of stimulus values is constant and the true value of the stimulus is $s = 0$. The dashed curve is for a Gaussian prior distribution with a mean of -2 and variance of 1, again with the true stimulus being $s = 0$. The peaks of the solid and dashed curves are at $s = 0.0385$ and $s = -0.107$, respectively.

Note that the bias depends on the true value of the stimulus. An estimate is termed unbiased if $b_{\text{est}}(s) = 0$ for all stimulus values.

variance

The variance of the estimator, which quantifies how much the estimate varies about its mean value, is defined as

$$\sigma^2_{\text{est}}(s) = \langle (s_{\text{est}} - \langle s_{\text{est}} \rangle)^2 \rangle \, . \tag{3.39}$$

estimation error

The bias and variance can be used to compute the trial-average squared estimation error, $\left\langle (s_{\text{est}} - s)^2 \right\rangle$. This is a measure of the spread of the estimated values about the true value of the stimulus. Because $s = \langle s_{\text{est}} \rangle - b_{\text{est}}(s)$, we can write the squared estimation error as

$$\left\langle (s_{\text{est}} - s)^2 \right\rangle = \left\langle (s_{\text{est}} - \langle s_{\text{est}} \rangle + b_{\text{est}}(s))^2 \right\rangle = \sigma^2_{\text{est}}(s) + b^2_{\text{est}}(s) \, . \tag{3.40}$$

In other words, the average squared estimation error is the sum of the variance and the square of the bias. For an unbiased estimate, the average squared estimation error is equal to the variance of the estimator.

Fisher Information

Decoding can be used to limit the accuracy with which a neural system encodes the value of a stimulus parameter because the encoding accuracy cannot exceed the accuracy of an optimal decoding method. Of course, we must be sure that the decoding technique used to establish such a bound is truly optimal, or else the result will reflect the limitations of the decoding procedure, not bounds on the neural system being studied. The Fisher information is a quantity that provides one such measure of encoding accuracy. Through a bound known as the Cramér-Rao bound, the Fisher information limits the accuracy with which any decoding scheme can extract an estimate of an encoded quantity.

Cramér-Rao bound

The Cramér-Rao bound limits the variance of any estimate s_{est} according

This is just a discrete time version of the sort of linear filter used in chapters 1 and 2.

Arranging for $v(t)$ to predict the total future reward would appear to require a simple modification of the delta rule we have discussed previously,

$$w(\tau) \rightarrow w(\tau) + \epsilon\delta(t)u(t-\tau)\,, \tag{9.7}$$

with $\delta(t)$ being the difference between the actual and predicted total future reward, $\delta(t) = \sum_\tau r(t+\tau) - v(t)$. However, there is a problem with applying this rule in a stochastic gradient descent algorithm. Computation of $\delta(t)$ requires knowledge of the total future reward on a given trial. Although $r(t)$ is known at time t, the succeeding $r(t+1), r(t+2)\ldots$ have yet to be experienced, making it impossible to calculate $\delta(t)$. A possible solution is suggested by the recursive formula

$$\sum_{\tau=0}^{T-t} r(t+\tau) = r(t) + \sum_{\tau=0}^{T-t-1} r(t+1+\tau)\,. \tag{9.8}$$

The temporal difference model of prediction is based on the observation that $v(t+1)$ provides an approximation of the average value (across trials) of the last term in equation 9.8, so we can write

$$\sum_{\tau=0}^{T-t} r(t+\tau) \approx r(t) + v(t+1)\,. \tag{9.9}$$

Replacing the sum in the equation $\delta(t) = \sum_\tau r(t+\tau) - v(t)$ by this approximation gives the temporal difference learning rule,

temporal difference rule

$$w(\tau) \rightarrow w(\tau) + \epsilon\delta(t)u(t-\tau) \quad \text{with} \quad \delta(t) = r(t) + v(t+1) - v(t)\,. \tag{9.10}$$

The name of the rule comes from the term $v(t+1) - v(t)$, which is the difference between two successive estimates. $\delta(t)$ is usually called the temporal difference error. Under a variety of circumstances, this rule is likely to converge to make the correct predictions.

Figure 9.2 shows what happens when the temporal difference rule is applied during a training period in which a stimulus appears at time $t = 100$, and a reward is given for a short interval around $t = 200$. Initially, $w(\tau) = 0$ for all τ. Figure 9.2A shows that the temporal difference error starts off being nonzero only at the time of the reward, $t = 200$, and then, over trials, moves backward in time, eventually stabilizing around the time of the stimulus, where it takes the value 2. This is equal to the (integrated) total reward provided over the course of each trial. Figure 9.2B shows the behavior during a trial of a number of variables before and after learning. After learning, the prediction $v(t)$ is 2 from the time the stimulus is first presented ($t = 100$) until the time the reward starts to be delivered. Thus, the temporal difference prediction error (δ) has a spike at $t = 99$. This spike persists, because $u(t) = 0$ for $t < 100$. The temporal difference term (Δv) is negative around $t = 200$, exactly compensating for the delivery of reward, and thus making $\delta = 0$.

important because the likelihood is expected to be at a maximum near the true stimulus value s that caused the responses. If the likelihood is very curved, and thus the Fisher information is large, responses typical for the stimulus s are much less likely to occur for slightly different stimuli. Therefore, the typical response provides a strong indication of the value of the stimulus. If the likelihood is fairly flat, and thus the Fisher information is small, responses common for s are likely to occur for slightly different stimuli as well. Thus, the response does not as clearly determine the stimulus value. The Fisher information is purely local in the sense that it does not reflect the existence of stimulus values completely different from s that are likely to evoke the same responses as those evoked by s itself. However, this does not happen for the sort of simple population codes we consider. Shannon's mutual information measure, discussed in chapter 4, takes such possibilities into account.

The Fisher information for a population of neurons with uniformly arrayed tuning curves (the Gaussian array in figure 3.8, for example) and Poisson statistics can be computed from the conditional firing-rate probability in equation 3.30. Because the spike-count rate is described here by a probability rather than a probability density, we use the discrete analog of equation 3.42,

$$I_{\rm F}(s) = \left\langle -\frac{\partial^2 \ln P[\mathbf{r}|s]}{\partial s^2} \right\rangle = T \sum_{a=1}^{N} \left(\langle r_a \rangle \left(\left(\frac{f_a'(s)}{f_a(s)} \right)^2 - \frac{f_a''(s)}{f_a(s)} \right) + f_a''(s) \right). \tag{3.44}$$

Note that we have used the full expression, equation 3.30, in deriving this result, not the truncated form of $\ln P[\mathbf{r}|s]$ in equation 3.31. We next make the replacement $\langle r_a \rangle = f_a(s)$, producing the final result

$$I_{\rm F}(s) = T \sum_{a=1}^{N} \frac{\left(f_a'(s)\right)^2}{f_a(s)}. \tag{3.45}$$

In this expression, each neuron contributes an amount to the Fisher information proportional to the square of its tuning curve slope and inversely proportional to the average firing rate for the particular stimulus value being estimated. Highly sloped tuning curves give firing rates that are sensitive to the precise value of the stimulus. Figure 3.11 shows the contribution to the sum in equation 3.45 from a single neuron with a Gaussian tuning curve, the neuron with $s_a = 0$ in figure 3.8. For comparison purposes, a dashed curve proportional to the tuning curve is also plotted. Note that the Fisher information vanishes for the stimulus value that produces the maximum average firing rate, because $f_a'(s) = 0$ at this point. The firing rate of a neuron at the peak of its tuning curve is relatively unaffected by small changes in the stimulus. Individual neurons carry the most Fisher information in regions of their tuning curves where average firing rates are rapidly varying functions of the stimulus value, not where the firing rate is highest.

The Fisher information can be used to derive an interesting result on the optimal widths of response tuning curves. Consider a population of neu-

paradigm that first led to the suggestion of the delta rule in connection with classical conditioning. In blocking, two stimuli are presented together with the reward, but only after an association has already developed for one stimulus by itself. In other words, during the pre-training period, a stimulus is associated with a reward, as in Pavlovian conditioning. Then, during the training period, a second stimulus is present along with the first, in association with the same reward. In this case, the preexisting association of the first stimulus with the reward blocks an association from forming between the second stimulus and the reward. Thus, after training, a conditioned response is evoked only by the first stimulus, not by the second. This follows from the vector form of the delta rule, because training with the first stimulus makes $w_1 = r$. When the second stimulus is presented along with the first, its weight starts out at $w_2 = 0$, but the prediction of reward $v = w_1 u_1 + w_2 u_2$ is still equal to r. This makes $\delta = 0$, so no further weight modification occurs.

inhibitory conditioning

A standard way to induce inhibitory conditioning is to use trials in which one stimulus is shown in conjunction with the reward in alternation with trials in which that stimulus and an additional stimulus are presented in the absence of reward. In this case, the second stimulus becomes a conditioned inhibitor, predicting the absence of reward. This can be demonstrated by presenting a third stimulus that also predicts reward, in conjunction with the inhibitory stimulus, and showing that the net prediction of reward is reduced. It can also be demonstrated by showing that subsequent learning of a positive association between the inhibitory stimulus and reward is slowed. Inhibition emerges naturally from the delta rule. Trials in which the first stimulus is associated with a reward result in a positive value of w_1. Over trials in which both stimuli are presented together, the net prediction $v = w_1 + w_2$ comes to be 0, so w_2 is forced to be negative.

overshadowing

A further example of the interaction between stimuli is overshadowing. If two stimuli are presented together during training, the prediction of reward is shared between them. After application of the delta rule, $v = w_1 + w_2 = r$. However, the prediction is often shared unequally, as if one stimulus is more salient than the other. Overshadowing can be encompassed by generalizing the delta rule so that the two stimuli have different learning rates (different values of ϵ), reflecting unequal associabilities. Weight modification stops when $\langle \delta \rangle = 0$, at which point the faster growing weight will be larger than the slower growing weight. Various, more subtle effects come from having not only different but also modifiable learning rates, but these lie beyond the scope of our account.

The Rescorla-Wagner rule, binary stimulus parameters, and linear reward prediction are obviously gross simplifications of animal learning behavior. Yet they summarize and unify an impressive amount of classical conditioning data and are useful, provided their shortcomings are fully appreciated. As a reminder of this, we point out one experiment, secondary conditioning, that cannot be encompassed within this scheme.

information is proportional to this number divided by the square of the tuning curve width. Combining these factors, the Fisher information is inversely proportional to σ_r, and the encoding accuracy increases with narrower tuning curves.

The advantage of using narrow tuning curves goes away if the stimulus is characterized by more than one parameter. Consider a stimulus with D parameters and suppose that the response tuning curves are products of identical Gaussians for each of these parameters. If the tuning curves cover the D-dimensional space of stimulus values with a uniform density ρ_s, the number of responding neurons for any stimulus value is proportional to $\rho_s \sigma_r^D$ and, using the same integral approximation as in equation 3.47, the Fisher information is

$$I_{\rm F} = \frac{(2\pi)^{D/2}\rho_s\sigma_r^D r_{\max} T}{D\sigma_r^2} = \frac{(2\pi)^{D/2}\rho_s\sigma_r^{D-2} r_{\max} T}{D}\,. \tag{3.48}$$

This equation, which reduces to the result given above if $D = 1$, allows us to examine the effect of tuning curve width on encoding accuracy. The trade-off between the encoding accuracy of individual neurons and the number of responding neurons depends on the dimension of the stimulus space. Narrowing the tuning curves (making σ_r smaller) increases the Fisher information for $D = 1$, decreases it for $D > 2$, and has no impact if $D = 2$.

Optimal Discrimination

In the first part of this chapter, we considered discrimination between two values of a stimulus. An alternative to the procedures discussed there is simply to decode the responses and discriminate on the basis of the estimated stimulus values. Consider the case of discriminating between s and $s + \Delta s$ for small Δs. For large N, the average value of the difference between the ML estimates for the two stimulus values is equal to Δs (because the estimate is unbiased) and the variance of each estimate (for small Δs) is $1/I_{\rm F}(s)$. Thus, the discriminability, defined in equation 3.4, for the ML-based test is

ML discriminability

$$d' = \Delta s\sqrt{I_{\rm F}(s)}\,. \tag{3.49}$$

The larger the Fisher information, the higher the discriminability. We leave as an exercise the proof that for small Δs, this discriminability is the same as that of the likelihood ratio test $Z(\mathbf{r})$ defined in equation 3.19.

Discrimination by ML estimation requires maximizing the likelihood, and this may be computationally challenging. The likelihood ratio test described previously may be simpler, especially for Poisson variability, because, for small Δs, the likelihood ratio test Z defined in equation 3.19 is a linear function of the firing rates,

$$Z = T\sum_{a=1}^{N} r_a \frac{f_a'(s)}{f_a(s)}\,. \tag{3.50}$$

Paradigm	Pre-Train	Train	Result
Pavlovian		$s \rightarrow r$	$s \rightarrow$ 'r'
Extinction	$s \rightarrow r$	$s \rightarrow \cdot$	$s \rightarrow$ '$\cdot$'
Partial		$s \rightarrow r \quad s \rightarrow \cdot$	$s \rightarrow \alpha$ 'r'
Blocking	$s_1 \rightarrow r$	$s_1 + s_2 \rightarrow r$	$s_1 \rightarrow$ 'r' $\quad s_2 \rightarrow$ '$\cdot$'
Inhibitory		$s_1 + s_2 \rightarrow \cdot \quad s_1 \rightarrow r$	$s_1 \rightarrow$ 'r' $\quad s_2 \rightarrow -$'r'
Overshadow		$s_1 + s_2 \rightarrow r$	$s_1 \rightarrow \alpha_1$'r' $\quad s_2 \rightarrow \alpha_2$'$r$'
Secondary	$s_1 \rightarrow r$	$s_2 \rightarrow s_1$	$s_2 \rightarrow$ 'r'

Table 9.1 Classical conditioning paradigms. The columns indicate the training procedures and results, with some paradigms requiring a pre-training as well as a training period. Both training and pre-training periods consist of a moderate number of training trials. The arrows represent an association between one or two stimuli (s, or s_1 and s_2) and either a reward (r) or the absence of a reward ($\cdot$). In Partial and Inhibitory conditioning, the two types of training trials that are indicated are alternated. In the Result column, the arrows represent an association between a stimulus and the expectation of a reward ('r') or no reward ('$\cdot$'). The factors of α denote a partial or weakened expectation, and the minus sign indicates the suppression of an expectation of reward.

is present, $u = 0$ if it is absent). The expected reward, denoted by v, is expressed as this stimulus variable multiplied by a weight w,

stimulus u
expected reward v
weight w

$$v = wu \; . \tag{9.1}$$

The value of the weight, w, is established by a learning rule designed to minimize the expected squared error between the actual reward r and the prediction v, $\langle (r - v)^2 \rangle$. The angle brackets indicate an average over the presentations of the stimulus and reward, either or both of which may be stochastic. As we saw in chapter 8, stochastic gradient descent in the form of the delta rule is one way of minimizing this error. This results in the trial-by-trial learning rule known as the Rescorla-Wagner rule,

Rescorla-Wagner rule

$$w \rightarrow w + \epsilon \delta u \quad \text{with} \quad \delta = r - v \; . \tag{9.2}$$

Here ϵ is the learning rate, which can be interpreted in psychological terms as the associability of the stimulus with the reward. The crucial term in this learning rule is the prediction error, δ. In a later section, we interpret the activity of dopaminergic cells in the ventral tegmental area (VTA) as encoding a form of this prediction error. If ϵ is sufficiently small and $u = 1$ on every trial (the stimulus is always presented), the rule ultimately makes w fluctuate about the equilibrium value $w = \langle r \rangle$, at which point the average value of δ is 0.

associability

The filled circles in figure 9.1 show how learning progresses according to the Rescorla-Wagner rule during the acquisition and extinction phases of Pavlovian conditioning. In this example, the stimulus and reward were both initially presented on each trial, but later the reward was removed. The weight approaches the asymptotic limit $w = r$ exponentially during the rewarded phase of training (conditioning), and exponentially decays to $w = 0$ during the unrewarded phase (extinction). Experimental learning

acquisition
extinction

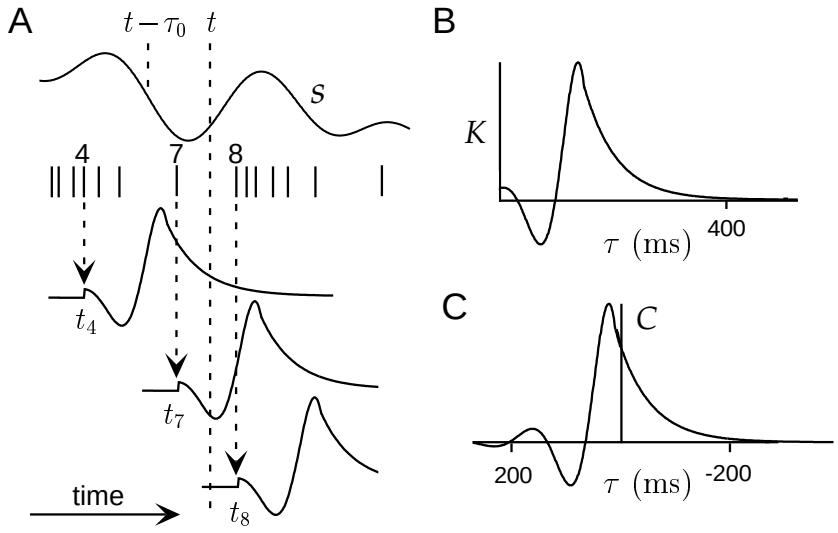

Figure 3.13 Illustration of spike-train decoding. (A) The top trace denotes a stimulus that evokes the spike train appearing below it (second trace from top). At time t an estimate is being made of the stimulus at time $t - \tau_0$. The estimate is obtained by summing the values of the kernels where they cross the dashed line labeled t, for spikes up to and including spike 7. Two such kernels are shown in the third and fourth traces from the top. The real estimate is obtained by summing similar contributions from all of the spikes. The kernel is 0 for negative values of its argument, so spikes for $i \geq 8$ do not contribute to the estimate at this time (e.g., fifth trace from top). (B) The kernel used in A. This has been truncated to zero value for negative values of τ. (C) The spike-triggered average corresponding to the kernel in B, assuming no spike-train correlations. Note that C has been plotted with the τ axis reversed, following the convention established in chapter 1. With this convention, K in panel B is simply a shifted and truncated version of the curve appearing here. In this case $\tau_0 = 160$ ms.

action potential about the value of the stimulus at a later time $t > t_i$. That is, the evoked spikes tell us about the past behavior of the stimulus and, in spike decoding, we attempt to use this information to predict the current stimulus value. Clearly, this requires that the stimulus have some form of temporal correlation so that past behavior provides information about the current stimulus value. To make the decoding task easier, we

prediction delay τ_0

can introduce a prediction delay, τ_0, and attempt to construct, from spikes occurring prior to time t, an estimate of the stimulus at time $t - \tau_0$ (see figure 3.13A). Such a delayed estimate uses a combination of spikes that could have been fired in response to the stimulus $s(t - \tau_0)$ being estimated (those for which $t - \tau_0 < t_i < t$; spike 7 in figure 3.13A), and spikes that occurred too early to be affected by the value of $s(t - \tau_0)$ (those for which $t_i < t - \tau_0$; spikes 1-6 in figure 3.13A), but that can contribute to its estimation on the basis of stimulus correlations. The estimation task gets easier as τ_0 is increased, but this delays the decoding and makes the result less behaviorally relevant. We will consider decoding with an arbitrary delay and later discuss how to set a specific value for τ_0.

stimulus estimate

The stimulus estimate is constructed as a linear sum over all spikes. A

9 Classical Conditioning and Reinforcement Learning

9.1 Introduction

The ability of animals to learn appropriate actions in response to particular stimuli on the basis of associated rewards or punishments is a focus of behavioral psychology. The field is traditionally separated into classical (or Pavlovian) and instrumental (or operant) conditioning. In classical conditioning, the reinforcers (i.e., the rewards or punishments) are delivered independently of any actions taken by the animal. In instrumental conditioning, the actions of the animal determine what reinforcement is provided. Learning about stimuli or actions solely on the basis of the rewards and punishments associated with them is called reinforcement learning. Reinforcement learning is minimally supervised because animals are not told explicitly what actions to take in particular situations, but must work this out for themselves on the basis of the reinforcement they receive.

classical and instrumental conditioning

reinforcement learning

We begin this chapter with a discussion of aspects of classical conditioning and the models that have been developed to account for them. We first discuss various pairings of one or more stimuli with presentation or denial of a reward, and present a simple learning algorithm that summarizes the results. We then present an algorithm, called temporal difference learning, that leads to predictions of both the presence and the timing of rewards delivered after a delay following stimulus presentation. Two neural systems, the cerebellum and the midbrain dopamine system, have been particularly well studied from the perspective of conditioning. The cerebellum has been studied in association with eyeblink conditioning, a paradigm in which animals learn to shut their eyes just in advance of disturbances, such as puffs of air, that are signaled by cues. The midbrain dopaminergic system has been studied in association with reward learning. We focus on the latter, together with a small fraction of the extensive behavioral data on conditioning.

There are two broad classes of instrumental conditioning tasks. In the first class, which we illustrate with an example of foraging by bees, the reinforcement is delivered immediately after the action is taken. This makes learning relatively easy. In the second class, the reward or punishment depends on an entire sequence of actions and is partly or wholly delayed

integral equation similar to 3.54 would be simplified. This could always be done because we have complete control over the stimulus in this type of experiment. However, we do not have similar control of the neuron, and must deal with whatever spike-train autocorrelation function it gives us. If the spike train is uncorrelated, which tends to happen at low rates,

$$Q_{\rho\rho}(\tau) = \langle r \rangle \delta(\tau) \,, \tag{3.57}$$

and we find from equation 3.54 that

$$K(\tau) = \frac{1}{\langle r \rangle} Q_{\rm rs}(\tau - \tau_0) = C(\tau_0 - \tau) = \frac{1}{\langle n \rangle} \left\langle \sum_{i=1}^{n} s(t_i + \tau - \tau_0) \right\rangle . \tag{3.58}$$

This is the average value of the stimulus at time $\tau - \tau_0$ relative to the appearance of a spike. Because $\tau - \tau_0$ can be either positive or negative, stimulus estimation, unlike firing-rate estimation, involves both forward and backward correlation and the average values of the stimulus both before and after a spike. Decoding in this way follows a simple rule: every time a spike appears, we replace it with the average stimulus surrounding a spike, shifted by an amount τ_0 (figure 3.13).

The need for either stimulus correlations or a nonzero prediction delay is clear from equation 3.58. Correlations between a spike and subsequent stimuli can arise, in a causal system, only from correlations between the stimulus and itself. If these are absent, as for white noise, $K(\tau)$ will be 0 for $\tau > \tau_0$. For causal decoding, we must also have $K(\tau) = 0$ for $\tau < 0$. Thus, if $\tau_0 = 0$ and the stimulus is uncorrelated, $K(\tau) = 0$ for all values of τ.

optimal kernel

When the spike-train autocorrelation function is not a δ function, an acausal solution for K can be expressed as an inverse Fourier transform,

$$K(\tau) = \frac{1}{2\pi} \int d\omega \, \tilde{K}(\omega) \exp(-i\omega\tau) \,, \tag{3.59}$$

where, as shown in appendix C,

$$\tilde{K}(\omega) = \frac{\tilde{Q}_{\rm rs}(\omega) \exp(i\omega\tau_0)}{\tilde{Q}_{\rho\rho}(\omega)} \,. \tag{3.60}$$

Here $\tilde{Q}_{\rm rs}$ and $\tilde{Q}_{\rho\rho}$ are the Fourier transforms of Q_{rs} and $Q_{\rho\rho}$. The numerator in this expression reproduces the expression $Q_{\rm rs}(\tau - \tau_0)$ in equation 3.58. The role of the denominator is to correct for any autocorrelations in the response spike train. Such correlations introduce a bias in the decoding, and the denominator in equation 3.60 corrects for this bias.

If we ignore the constraint of causality, then, because the occurrence of a spike cannot depend on the behavior of a stimulus in the very distant past, we can expect $K(\tau)$ from equation 3.58 to vanish for sufficiently negative values of $\tau - \tau_0$. For most neurons, this will occur for $\tau - \tau_0$ more negative than minus a few hundred ms. The decoding kernel can therefore be made

1996; Kempter et al., 1999; Song et al., 2000). Plasticity of intrinsic conductance properties of neurons, as opposed to synaptic plasticity, is considered in LeMasson et al. (1993), Liu et al. (1999), and Stemmler & Koch (1999).

Descriptions of relevant data on the patterns of responsivity across cortical areas and the development of these patterns include Hubener et al. (1997), **Yuste & Sur (1999)**, and Weliky (2000). **Price & Willshaw (2000)** offers a broad-based, theoretically informed review. Recent experimental challenges to plasticity-based models include Crair et al. (1998) and Crowley & Katz (1999). Neural pattern formation mechanisms involving chemical matching, which are likely important at least for establishing coarse maps, are reviewed from a theoretical perspective in **Goodhill & Richards (1999)**. The use of learning algorithms to account for cortical maps is reviewed in **Erwin et al. (1995)**, **Miller (1996a)**, and **Swindale (1996)**. The underlying mathematical basis of some rules is closely related to the reaction diffusion theory of morphogenesis of **Turing (1952)**. Other rules are motivated on the basis of minimizing quantities such as wire length in cortex.

We described Hebbian models for the development of ocular dominance and orientation selectivity due to Linsker (1986), Miller et al. (1989), and Miller (1994); a competitive Hebbian model closely related to that of Goodhill (1993) and Piepenbrock & Obermayer (1999); a self-organizing map model related to that of Obermayer et al. (1992); and the elastic net (Durbin & Willshaw, 1987) model of Durbin & Mitchison (1990), Goodhill & Willshaw (1990), and **Erwin et al. (1995)**. The first feature-based models were called noise models (see **Swindale, 1996**).

The capacity of a perceptron for random associations is computed in Cover (1965) and Venkatesh (1986). The perceptron learning rule is due to Rosenblatt (1958; see also **Minsky & Papert, 1969**). The delta rule was introduced by Widrow & Hoff (1960; see also Widrow & Stearns, 1985) and independently arose in other fields. The widely used backpropagation algorithm (see Chauvin & Rumelhart, 1995) is a form of delta rule learning that works in a larger class of networks. O'Reilly (1996) suggests a more biologically plausible implementation.

Supervised learning for classification and function approximation, and its ties to Bayesian and frequentist statistical theory, are reviewed in **Duda & Hart (1973)**, **Duda at al. (2000)**, **Kearns & Vazirani (1994)**, and **Bishop (1995)**. Poggio and colleagues have explored basis function models of various representational and learning phenomena (see Poggio, 1990). Tight frames are discussed in Daubechies et al. (1986) and are applied to visual receptive fields by Salinas & Abbott (2000).

Contrastive Hebbian learning is due to Hinton & Sejnowski (1986). See Hinton (2000) for discussion of the Boltzmann machine without recurrent connections, and for an alternative learning rule.

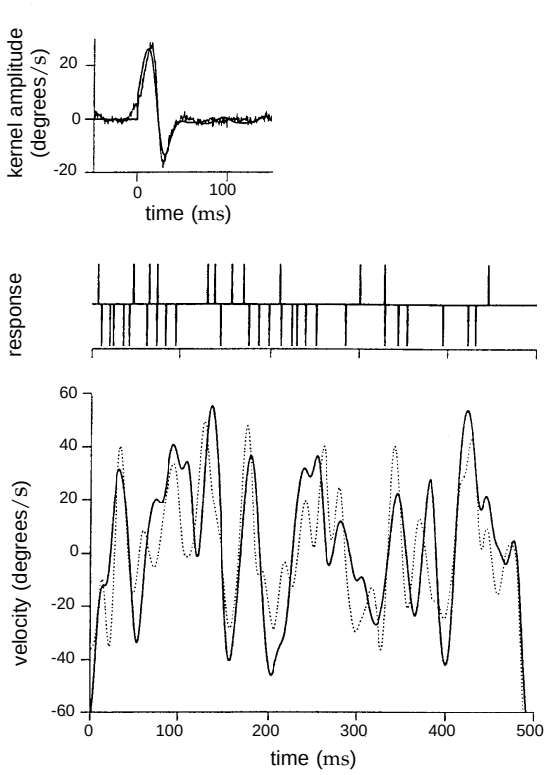

Figure 3.14 Decoding the stimulus from an H1 neuron of the fly. The upper panel is the decoding kernel. The jagged curve is the optimal acausal filter, and the smooth curve is a kernel obtained by expanding in a causal set of basis functions. In both cases, the kernels are shifted by $\tau_0 = 40$ ms. The middle panel shows typical responses of the H1 neuron to the stimuli $s(t)$ (upper trace) and $-s(t)$ (bottom trace). The dashed line in the lower panel shows the actual stimulus, and the solid line is the estimated stimulus from the optimal linear reconstruction using the acausal filter. (Adapted from Rieke et al., 1997.)

from a population of M1 neurons, the stimuli used are extremely simple compared with the naturally occurring stimuli that must be interpreted during normal behavior.

3.5 Chapter Summary

We have considered the decoding of stimulus characteristics from the responses they evoke, including discrimination between stimulus values, the decoding of static stimuli on the basis of population responses, and the decoding of dynamic stimulus parameters from spike trains. Discrimination was studied using the receiver operating characteristic, likelihood ra-

plastic. In the latter case, anti-Hebbian plasticity can ensure decorrelation of multiple output units. We also considered the role of competition and cooperation in models of activity-dependent development, and described two examples of feature-based models, the self-organizing map and the elastic net.

Finally, we considered supervised learning applied to binary classification and function approximation, using supervised Hebbian learning, the perceptron learning rule, and gradient descent learning through the delta rule. We also treated contrastive Hebbian learning for the Boltzmann machine, involving Hebbian and anti-Hebbian updates in different phases.

8.6 Appendix

Convergence of the Perceptron Learning Rule

For convenience, we take $\epsilon_w = 1$ and start the perceptron learning rule (equation 8.56) with $\mathbf{w} = \mathbf{0}$ and $\gamma = 0$. Then, under presentation of the sample m, the changes in the weights and threshold are given by

$$\Delta \mathbf{w} = \frac{1}{2}(v^m - v(\mathbf{u}^m))\mathbf{u}^m \quad \text{and} \quad \Delta\gamma = -\frac{1}{2}(v^m - v(\mathbf{u}^m)) \,. \tag{8.73}$$

Given a finite, linearly separable problem, there must be a set of weights $\mathbf{w}^*$ and a threshold γ^* that are normalized ($|\mathbf{w}^*|^2 + (\gamma^*)^2 = 1$) and allow the perceptron to categorize correctly, for which we require the condition $(\mathbf{w}^* \cdot \mathbf{u}^m - \gamma^*)v^m > \delta$ for some $\delta > 0$ and for all m.

Consider the cosine of the angle between the current weights and threshold $\mathbf{w}$, γ and the solution $\mathbf{w}^*$, γ^*

$$\Phi(\mathbf{w}, \gamma) = \frac{\mathbf{w} \cdot \mathbf{w}^* + \gamma\gamma^*}{\sqrt{|\mathbf{w}|^2 + (\gamma)^2}} = \frac{\psi(\mathbf{w}, \gamma)}{|\mathbf{w}, \gamma|} \,, \tag{8.74}$$

to introduce some compact notation. The perceptron convergence theorem proves that the perceptron learning rule solves any solvable categorization problem, because assuming otherwise would imply that Φ would eventually grow larger than 1. This is impossible for a cosine function, which must lie between -1 and 1.

To show this, we consider the change in ψ due to one step of perceptron learning during which $\mathbf{w}$ and γ are modified because the current weights generated the wrong response. When an incorrect response is generated, $v(\mathbf{u}^m) = -v^m$, so $(v^m - v(\mathbf{u}^m))/2 = v^m$, and thus

$$\Delta\psi = (\mathbf{w}^* \cdot \mathbf{u}^m - \gamma^*)v^m > \delta \,. \tag{8.75}$$

The inequality follows from the condition imposed on $\mathbf{w}^*$ and γ^* as providing a solution of the categorization problem. Assuming that ψ is initially positive and iterating this result over n steps in which the weights

Putting back the region of integration that cancels between these two terms (for which $l(r) \geq z_l$ and $h(r) \geq z_h$), we find

$$\Delta\beta \geq z\left[\int dr\, p[r|-]\Theta(l(r)-z_l) - \int dr\, p[r|-]\Theta(h(r)-z_h)\right]. \quad (3.66)$$

By definition, these integrals are the sizes of the two tests, which are equal by hypothesis. Thus $\Delta\beta \geq 0$, showing that no test can be better than the likelihood ratio $l(r)$, at least in the sense of maximizing the power for a given size.

B: The Cramér-Rao Bound

Cauchy-Schwarz inequality

The Cramér-Rao lower bound for an estimator s_{est} is based on the Cauchy-Schwarz inequality, which states that for any two quantities A and B,

$$\langle A^2\rangle\langle B^2\rangle \geq \langle AB\rangle^2\,. \quad (3.67)$$

To prove this inequality, note that

$$\left\langle \left(\langle B^2\rangle A - \langle AB\rangle B\right)^2 \right\rangle \geq 0 \quad (3.68)$$

because it is the average value of a square. Computing the square gives

$$\langle B^2\rangle^2\langle A^2\rangle - \langle AB\rangle^2\langle B^2\rangle \geq 0\,, \quad (3.69)$$

from which the inequality follows directly.

Consider the inequality of equation 3.67 with $A = \partial \ln p/\partial s$ and $B = s_{\text{est}} - \langle s_{\text{est}}\rangle$. From equations 3.43 and 3.39, we have $\langle A^2\rangle = I_{\text{F}}$ and $\langle B^2\rangle = \sigma^2_{\text{est}}$. The Cauchy-Schwarz inequality then gives

$$\sigma^2_{\text{est}}(s) I_{\text{F}} \geq \left\langle \frac{\partial \ln p[\mathbf{r}|s]}{\partial s}(s_{\text{est}} - \langle s_{\text{est}}\rangle)\right\rangle^2. \quad (3.70)$$

To evaluate the expression on the right side of the inequality 3.70, we differentiate the defining equation for the bias (equation 3.38),

$$s + b_{\text{est}}(s) = \langle s_{\text{est}}\rangle = \int d\mathbf{r}\, p[\mathbf{r}|s] s_{\text{est}}\,, \quad (3.71)$$

with respect to s to obtain

$$\begin{aligned} 1 + b'_{\text{est}}(s) &= \int d\mathbf{r}\, \frac{\partial p[\mathbf{r}|s]}{\partial s} s_{\text{est}} \\ &= \int d\mathbf{r}\, p[\mathbf{r}|s] \frac{\partial \ln p[\mathbf{r}|s]}{\partial s} s_{\text{est}} \\ &= \int d\mathbf{r}\, p[\mathbf{r}|s] \frac{\partial \ln p[\mathbf{r}|s]}{\partial s}(s_{\text{est}} - \langle s_{\text{est}}\rangle)\,. \end{aligned} \quad (3.72)$$

we discussed in chapter 7 to approximate the average over the distribution $P[\mathbf{v}|\mathbf{u}^m; \mathbf{W}]$ in equation 8.65.

Supervised learning can also be implemented in a Boltzmann machine with recurrent connections. When the output units are connected by a symmetric recurrent weight matrix $\mathbf{M}$ (with $M_{aa} = 0$), the energy function is

$$E(\mathbf{u}, \mathbf{v}) = -\mathbf{v} \cdot \mathbf{W} \cdot \mathbf{u} - \frac{1}{2}\mathbf{v} \cdot \mathbf{M} \cdot \mathbf{v} . \tag{8.67}$$

Everything that has been described thus far applies to this case, except that the output $\mathbf{v}(\mathbf{u}^m)$ for the sample input $\mathbf{u}^m$ must now be computed by repeated Gibbs sampling, using $F(\sum_b W_{ab} u_b^m + \sum_{a'} M_{aa'} v_{a'})$ for the probability that $v_a = 1$ (see chapter 7). Repeated sampling is required to assure that the network relaxes to the equilibrium distribution of equation 8.62. Modification of the feedforward weight W_{ab} then proceeds as in equation 8.66. The contrastive Hebb rule for recurrent weight $M_{aa'}$ is similarly given by

supervised learning for $\mathbf{M}$

$$M_{aa'} \rightarrow M_{aa'} + \epsilon_m \left(v_a^m v_{a'}^m - v_a(\mathbf{u}^m) v_{a'}(\mathbf{u}^m) \right) . \tag{8.68}$$

The Boltzmann machine was originally introduced in the context of unsupervised rather than supervised learning. In the supervised case, we try to make the distribution $P[\mathbf{v}|\mathbf{u}; \mathbf{W}]$ match the probability distribution $P[\mathbf{v}|\mathbf{u}]$ that generates the samples pairs $(\mathbf{u}^m, \mathbf{v}^m)$. In the unsupervised case, no output sample $\mathbf{v}^m$ is provided, and instead we try to make the network generate a probability distribution over $\mathbf{u}$ that matches the distribution $P[\mathbf{u}]$ from which the samples $\mathbf{u}^m$ were drawn. As we discuss in chapter 10, a frequent goal of probabilistic unsupervised learning is to generate network distributions that match the distributions of input data.

We consider the unsupervised Boltzmann machine without recurrent connections. In addition to the distribution of equation 8.62 for $\mathbf{v}$, given a specific input $\mathbf{u}$, the energy function of the Boltzmann machine can be used to define a distribution over both $\mathbf{u}$ and $\mathbf{v}$ defined by

$$P[\mathbf{u}, \mathbf{v}; \mathbf{W}] = \frac{\exp(-E(\mathbf{u}, \mathbf{v}))}{Z} \quad \text{with} \quad Z = \sum_{\mathbf{u},\mathbf{v}} \exp(-E(\mathbf{u}, \mathbf{v})) . \tag{8.69}$$

This can be used to construct a distribution for $\mathbf{u}$ alone by summing over the possible values of $\mathbf{v}$,

$$P[\mathbf{u}; \mathbf{W}] = \sum_{\mathbf{v}} P[\mathbf{u}, \mathbf{v}; \mathbf{W}] = \frac{1}{Z} \sum_{\mathbf{v}} \exp(-E(\mathbf{u}, \mathbf{v})) . \tag{8.70}$$

The goal of unsupervised learning for the Boltzmann machine is to make this distribution match, as closely as possible, the distribution of inputs P[$\mathbf{u}$].

3.7 Annotated Bibliography

Statistical analysis of discrimination, various forms of decoding, the Neyman-Pearson lemma, the Fisher information, and the Cramér-Rao lower bound can be found in **Cox & Hinckley (1974)**. Receiver operator characteristics and signal detection theory are described comprehensively in **Green & Swets (1966)** and Graham (1989). Our account of spike-train decoding follows that of **Rieke et al. (1997)**. Spectral factorization is discussed in Poor (1994). **Newsome et al. (1989)** and **Salzman et al. (1992)** present important results concerning visual motion discrimination and recordings from area MT, and **Shadlen et al. (1996)** provides a theoretically oriented review.

The vector method of population decoding has been considered in the context of a number of systems, and references include Humphrey et al. (1970), Georgopoulos et al. (1986 & 1988), van Gisbergen et al. (1987), and Lee et al. (1988). Various theoretical aspects of population decoding, such as vector and ML decoding and the Fisher information, that comprise our account were developed by Paradiso (1988), Baldi and Heiligenberg (1988), Vogels (1990), Snippe & Koenderink (1992), Zohary (1992), Seung & Sompolinsky (1993), Touretzky et al. (1993), Salinas & Abbott (1994), Sanger (1994, 1996), Snippe (1996), and Oram et al. (1998). Population codes are also known as coarse codes in the connectionist literature (Hinton, 1981). In our discussion of the effect of tuning curve widths on the Fisher information, we followed Zhang and Sejnowski (1999), but see also Snippe & Koenderink (1992) and Hinton (1984).

procedure gives rise to a conditional probability distribution $P[\mathbf{v}|\mathbf{u};\mathbf{W}]$ for $\mathbf{v}$ given $\mathbf{u}$ that can be written as

$$P[\mathbf{v}|\mathbf{u};\mathbf{W}] = \frac{\exp(-E(\mathbf{u},\mathbf{v}))}{Z(\mathbf{u})} \quad \text{with} \quad Z(\mathbf{u}) = \sum_{\mathbf{v}} \exp(-E(\mathbf{u},\mathbf{v}))\,, \tag{8.62}$$

where $E(\mathbf{u},\mathbf{v}) = -\mathbf{v}\cdot\mathbf{W}\cdot\mathbf{u}$. In this case, the partition function can be written as $Z(\mathbf{u}) = \prod_a (1+\exp(\sum_b W_{ab}u_b))$. However, for the Boltzmann machines with recurrent connections that we consider below, there is no simple closed form expression for the partition function.

The natural measure for determining how well the distribution generated by the network, $P[\mathbf{v}|\mathbf{u};\mathbf{W}]$, matches the sampled distribution, $P[\mathbf{v}|\mathbf{u}]$, for a particular input $\mathbf{u}$ is the Kullback-Leibler divergence,

$$\begin{aligned} D_{\mathrm{KL}}(P[\mathbf{v}|\mathbf{u}],\, P[\mathbf{v}|\mathbf{u};\mathbf{W}]) &= \sum_{\mathbf{v}} P[\mathbf{v}|\mathbf{u}] \ln\left(\frac{P[\mathbf{v}|\mathbf{u}]}{P[\mathbf{v}|\mathbf{u};\mathbf{W}]}\right) \\ &= -\sum_{\mathbf{v}} P[\mathbf{v}|\mathbf{u}] \ln\left(P[\mathbf{v}|\mathbf{u};\mathbf{W}]\right) + K\,, \end{aligned} \tag{8.63}$$

where K is a term that is proportional to the entropy of the distribution $P[\mathbf{v}|\mathbf{u}]$ (see chapter 4). We do not write out this term explicitly because it does not depend on the feedforward weight matrix, so it does not affect the learning rule used to modify $\mathbf{W}$. As in chapter 7, we have, for convenience, used natural logarithms (rather than base 2 as in chapter 4) in the definition of the Kullback-Leibler divergence.

To estimate, from the samples, how well $P[\mathbf{v}|\mathbf{u};\mathbf{W}]$ matches $P[\mathbf{v}|\mathbf{u}]$ across the different values of $\mathbf{u}$, we average the Kullback-Leibler divergence over all the input samples $\mathbf{u}^m$. Furthermore, the sum over all $\mathbf{v}$ with weighting factor $P[\mathbf{v}|\mathbf{u}]$ in equation 8.63 can be replaced with an average over the sample pairs $(\mathbf{u}^m,\mathbf{v}^m)$ because $\mathbf{v}^m$ is chosen from the probability distribution $P[\mathbf{v}|\mathbf{u}]$. Using brackets to denote the average over samples, this results in the measure

$$\langle D_{\mathrm{KL}}(P[\mathbf{v}|\mathbf{u}],\, P[\mathbf{v}|\mathbf{u};\mathbf{W}])\rangle = -\frac{1}{N_{\mathrm{S}}}\sum_{m=1}^{N_{\mathrm{S}}} \ln\left(P[\mathbf{v}^m|\mathbf{u}^m;\mathbf{W}]\right) + \langle K\rangle \tag{8.64}$$

for comparing $P[\mathbf{v}|\mathbf{u};\mathbf{W}]$ and $P[\mathbf{v}|\mathbf{u}]$. Each logarithmic term in the sum on the right side of this equation is the logarithm of the probability that a sample output $\mathbf{v}^m$ would have been drawn from the distribution $P[\mathbf{v}|\mathbf{u}^m;\mathbf{W}]$, when in fact it was drawn from $P[\mathbf{v}|\mathbf{u}^m]$. This makes the sum in equation 8.64 equal to the logarithm of the likelihood that the sample data could have been produced from $P[\mathbf{v}|\mathbf{u}^m;\mathbf{W}]$. As a result, finding the network distribution $P[\mathbf{v}|\mathbf{u}^m;\mathbf{W}]$ that best matches $P[\mathbf{v}|\mathbf{u}^m]$ (in the sense of minimizing the Kullback-Leibler divergence) is equivalent to maximizing the likelihood that the sample $\mathbf{v}^m$ could have been drawn from $P[\mathbf{v}|\mathbf{u}^m;\mathbf{W}]$.

log likelihood

likelihood maximization

A learning rule that performs gradient ascent of the log likelihood can be derived by changing the weights by an amount proportional to the derivative of the logarithmic term in equation 8.64 with respect to the weights.

data. For this reason, many information theory analyses use simplified descriptions of the response of a neuron that reduce the number of possible "symbols" (i.e., responses) that need to be considered. We discuss cases in which the symbols consist of responses described by spike-count firing rates. We also consider the extension to continuous-valued firing rates. Because a reduced description of a spike train can carry no more information than the full spike train itself, this approach provides a lower bound on the actual information carried by the spike train.

Entropy

Entropy is a quantity that, roughly speaking, measures how "interesting" or "surprising" a set of responses is. Suppose that we are given a set of neural responses. If each response is identical, or if only a few different responses appear, we might conclude that this data set is relatively uninteresting. A more interesting set might show a larger range of different responses, perhaps in a highly irregular and unpredictable sequence. How can we quantify this intuitive notion of an interesting set of responses?

We begin by characterizing the responses in terms of their spike-count firing rates (i.e., the number of spikes divided by the trial duration), which can take a discrete set of different values. The methods we discuss are based on the probabilities $P[r]$ of observing a response with a spike-count rate r. The most widely used measure of entropy, due to Shannon, expresses the "surprise" associated with seeing a response rate r as a function of the probability of getting that response, $h(P[r])$, and quantifies the entropy as the average of $h(P[r])$ over all possible responses. The function $h(P[r])$, which acts as a measure of surprise, is chosen to satisfy a number of conditions. First, $h(P[r])$ should be a decreasing function of $P[r]$ because low probability responses are more surprising than high probability responses. Further, the surprise measure for a response that consists of two independent spike counts should be the sum of the measures for each spike count separately. This assures that the entropy and information measures we ultimately obtain will be additive for independent sources. Suppose we record rates r_1 and r_2 from two neurons that respond independently of each other. Because the responses are independent, the probability of getting this pair of responses is the product of their individual probabilities, $P[r_1]P[r_2]$, so the additivity condition requires that

$$h(P[r_1]P[r_2]) = h(P[r_1]) + h(P[r_2])\,. \tag{4.1}$$

The logarithm is the only function that satisfies such an identity for all P. Thus, it only remains to decide what base to use for the logarithm. By convention, base 2 logarithms are used so that information can be compared easily with results for binary systems. To indicate that the base 2 logarithm is being used, information is reported in units of "bits", with

bits

$$h(P[r]) = -\log_2 P[r]\,. \tag{4.2}$$

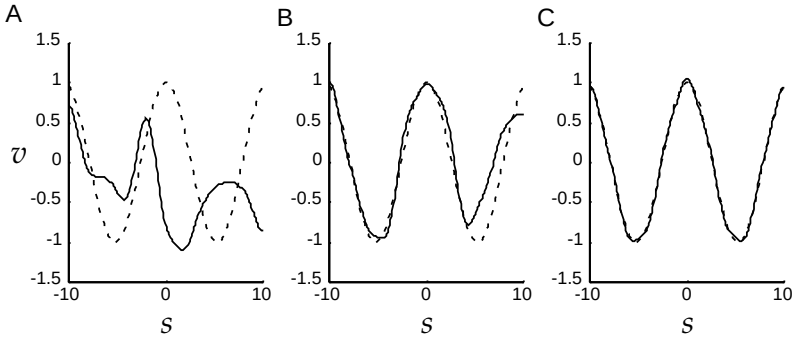

Figure 8.14 Eleven input neurons with Gaussian tuning curves drive an output neuron to approximate a sine function. The input tuning curves are $f_b(s) = \exp[-0.5(s - s_b)^2]$ with $s_b = -10, -8, -6, \ldots, 8, 10$. The delta rule was used to adjust the weights. Sample points were chosen randomly with s in the range between -10 and 10. The firing rate of the output neuron is plotted as a solid curve, and the sinusoidal target function as a dashed curve. (A) The firing rate of the output neuron when random weights in the range between -1 and 1 were used. (B) The output firing rate after weight modification using the delta rule with 20 sample points. (C) The output firing rate after weight modification using the delta rule with 100 sample points.

Gaussian tuning curves drives an output neuron so that it quite accurately represents a sine function. Figures 8.14B and C illustrate the difference between storage and generalization. The output $v(s)$ in figure 8.14B matches the sine function well for values of s that were in the training set, and, in this sense, has stored that information. However, it does not generalize well, in that it does not match the sine function for other values of s not in the training set. The output $v(s)$ in figure 8.14C has good storage and generalization properties, at least within the range of values of s used. The ability of the network to approximate the function $h(s)$ for stimulus values not presented during training depends in a complicated way on the smoothness of the target function, the number and smoothness of the basis functions $\mathbf{f}(s)$, and the size of the training set.

It is not immediately obvious how the delta rule of equation 8.61 could be implemented biophysically, because the network has to compute the difference $h(s^m)\mathbf{f}(s^m) - v(s^m)\mathbf{f}(s^m)$. One possibility is that the two terms $h(s^m)\mathbf{f}(s^m)$ and $v(s^m)\mathbf{f}(s^m)$ are computed in separate phases. First, the output of the network is clamped to the desired value $h(s^m)$ and Hebbian plasticity is applied. Then, the network runs freely to generate $v(s^m)$ and anti-Hebbian modifications are made. In the next section, we discuss a particular example of this in the case of the Boltzmann machine, and we show how learning rules intended for supervised learning can sometimes be used for unsupervised learning as well.

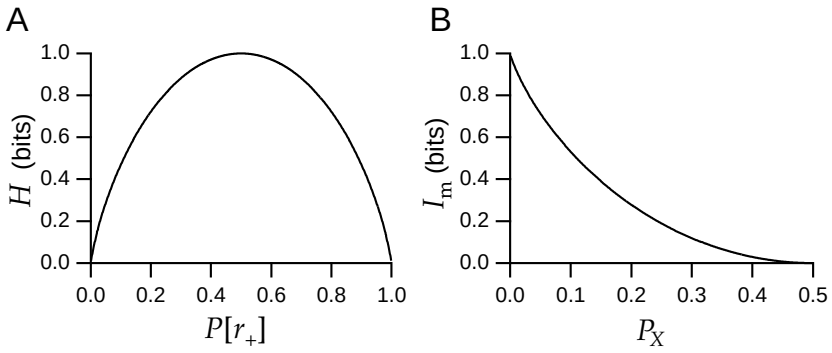

Figure 4.1 (A) The entropy of a binary code. $P[r_+]$ is the probability of a response at rate r_+, and $P[r_-] = 1 - P[r_+]$ is the probability of the other response, r_-. The entropy is maximum when $P[r_-] = P[r_+] = 1/2$. (B) The mutual information for a binary encoding of a binary stimulus. P_X is the probability of an incorrect response being evoked. The plot shows only $P_X \leq 1/2$ because values of $P_X > 1/2$ correspond to an encoding in which the relationship between the two responses and the two stimuli is reversed and the error probability is $1 - P_X$.

stimulus. Responses that are informative about the identity of the stimulus should exhibit larger variability for trials involving different stimuli than for trials that use the same stimulus repetitively. Mutual information is an entropy-based measure related to this idea.

The mutual information is the difference between the total response entropy and the average response entropy on trials that involve repetitive presentation of the same stimulus. Subtracting the entropy when the stimulus does not change removes from the total entropy the contribution from response variability that is not associated with the identity of the stimulus. When the responses are characterized by a spike-count rate, the total response entropy is given by equation 4.3. The entropy of the responses evoked by repeated presentations of a given stimulus s is computed using the conditional probability $P[r|s]$, the probability of a response at rate r given that stimulus s was presented, instead of the response probability $P[r]$ in equation 4.3. The entropy of the responses to a given stimulus is thus

$$H_s = -\sum_r P[r|s] \log_2 P[r|s] \,. \tag{4.5}$$

noise entropy

If we average this quantity over all the stimuli, we obtain a quantity called the noise entropy

$$H_{\text{noise}} = \sum_s P[s] H_s = -\sum_{s,r} P[s] P[r|s] \log_2 P[r|s] \,. \tag{4.6}$$

This is the entropy associated with that part of the response variability that is not due to changes in the stimulus, but arises from other sources. The mutual information is obtained by subtracting the noise entropy from the

alternative learning strategy is to start with an initial guess for the weights, compare the output $v(\mathbf{u}^m)$ in response to input $\mathbf{u}^m$ with the desired output v^m, and change the weights to improve the performance. Two important error-correcting modification rules are the perceptron rule, which applies to binary classification, and the delta rule, which can be applied to function approximation and many other problems.

The Perceptron Learning Rule

Suppose that the perceptron of equation 8.46 (with nonzero γ) incorrectly classifies an input pattern $\mathbf{u}^m$. If the output is $v(\mathbf{u}^m) = -1$ when $v^m = 1$, the weight vector should be modified to make $\mathbf{w} \cdot \mathbf{u}^m - \gamma$ larger. Similarly, if $v(\mathbf{u}^m) = 1$ when $v^m = -1$, $\mathbf{w} \cdot \mathbf{u}^m - \gamma$ should be decreased. A plasticity rule that performs such an adjustment is the perceptron learning rule,

perceptron learning rule

$$\mathbf{w} \to \mathbf{w} + \frac{\epsilon_w}{2}\left(v^m - v(\mathbf{u}^m)\right)\mathbf{u}^m \quad \text{and} \quad \gamma \to \gamma - \frac{\epsilon_w}{2}(v^m - v(\mathbf{u}^m)) \,. \tag{8.56}$$

Here, and in subsequent sections in this chapter, we use discrete updates for the weights (indicated by the $\to$) rather than the differential equations used up to this point. This is due to the discrete nature of the presentation of the training patterns. In equation 8.56, we have assumed that the threshold γ is also plastic. The learning rule for γ is inverted compared with that for the weights, because γ enters equation 8.46 with a minus sign.

To verify that the perceptron learning rule makes appropriate weight adjustments, we note that it implies that

$$\left(\mathbf{w} \cdot \mathbf{u}^m - \gamma\right) \to \left(\mathbf{w} \cdot \mathbf{u}^m - \gamma\right) + \frac{\epsilon_w}{2}(v^m - v(\mathbf{u}^m))\left(|\mathbf{u}^m|^2 + 1\right) \,. \tag{8.57}$$

This result shows that if $v^m = 1$ and $v(\mathbf{u}^m) = -1$, the weight change increases $\mathbf{w} \cdot \mathbf{u}^m - \gamma$. If $v^m = -1$ and $v(\mathbf{u}^m) = 1$, $\mathbf{w} \cdot \mathbf{u}^m - \gamma$ is decreased. This is exactly what is needed to compensate for the error. Note that the perceptron learning rule does not modify the weights if the output is correct.

To learn a set of input pattern classifications, the perceptron learning rule is applied to each one repeatedly, either sequentially or in a random order. For fixed ϵ_w, the perceptron learning rule of equation 8.56 is guaranteed to find a set of weights $\mathbf{w}$ and threshold γ that solve any linearly separable problem. This is proved in the appendix.

The Delta Rule

The function approximation task with the error function E of equation 8.52 can be solved using an error-correcting scheme similar in spirit to the perceptron learning rule, but designed for continuous rather than binary outputs. A simple but extremely useful version of this is the gradient descent procedure, which modifies $\mathbf{w}$ according to

gradient descent

distinct response r_s. Then, $P[r_s] = P[s]$ and $P[r|s]$ is 1 if $r = r_s$ and 0 otherwise. This causes the sum over r in equation 4.9 to collapse to just one term, and the mutual information becomes

$$I_{\rm m} = \sum_s P[s] \log_2 \left(\frac{1}{P[r_s]} \right) = -\sum_s P[s] \log_2 P[s] \,. \tag{4.13}$$

The last expression, which follows from the fact that $P[r_s] = P[s]$, is the entropy of the stimulus. Thus, with no variability and a one-to-one map from stimulus to response, the mutual information is equal to the full stimulus entropy.

Finally, imagine that there are only two possible stimulus values, which we label $+$ and $-$, and that the neuron responds with just two rates, r_+ and r_-. We associate the response r_+ with the $+$ stimulus, and the response r_- with the $-$ stimulus, but the encoding is not perfect. The probability of an incorrect response is $P_{\rm X}$, meaning that for the correct responses $P[r_+|+] = P[r_-|-] = 1 - P_{\rm X}$, and for the incorrect responses $P[r_+|-] = P[r_-|+] = P_{\rm X}$. We assume that the two stimuli are presented with equal probability so that $P[r_+] = P[r_-] = 1/2$, which, from equation 4.4, makes the full response entropy 1 bit. The noise entropy is $-(1 - P_{\rm X}) \log_2 (1 - P_{\rm X}) - P_{\rm X} \log_2 P_{\rm X}$. Thus, the mutual information is

$$I_{\rm m} = 1 + (1 - P_{\rm X}) \log_2 (1 - P_{\rm X}) + P_{\rm X} \log_2 P_{\rm X} \,. \tag{4.14}$$

This is plotted in figure 4.1B. When the encoding is error-free ($P_{\rm X} = 0$), the mutual information is 1 bit, which is equal to both the full response entropy and the stimulus entropy. When the encoding is random ($P_{\rm X} = 1/2$), the mutual information goes to 0.

It is instructive to consider this example from the perspective of decoding. We can think of the neuron as being a communication channel that reports noisily on the stimulus. From this perspective, we want to know the probability that a $+$ was presented, given that the response r_+ was recorded. By Bayes theorem, this is $P[+|r_+] = P[r_+|+]P[+]/P[r_+] = 1 - P_{\rm X}$. Before the response is recorded, the expectation was that $+$ and $-$ were equally likely. If the response r_+ is recorded, this expectation changes to $1 - P_{\rm X}$. The mutual information measures the corresponding reduction in uncertainty or, equivalently, the tightening of the posterior distribution due to the response.

KL divergence

The mutual information is related to a measure used in statistics called the Kullback-Leibler (KL) divergence. The KL divergence between one probability distribution $P[r]$ and another distribution $Q[r]$ is

$$D_{\rm KL}(P, Q) = \sum_r P[r] \log_2 \left(\frac{P[r]}{Q[r]} \right) . \tag{4.15}$$

The KL divergence has a property normally associated with a distance measure, $D_{\rm KL}(P, Q) \geq 0$ with equality if and only if $P = Q$ (proven in appendix A). However, unlike a distance, it is not symmetric with respect to

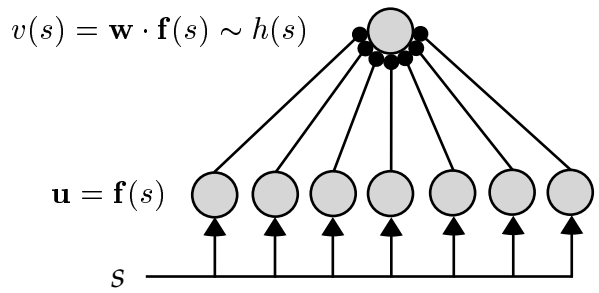

Figure 8.13 A network for representing functions. The value of an input variable s is encoded by the activity of a population of neurons with tuning curves $\mathbf{f}(s)$. This activity drives an output neuron through a vector of weights $\mathbf{w}$ to create an output activity v that approximates the function $h(s)$.

firing rate as representing the function. Populations of neurons (labeled by an index $b = 1, 2, \ldots, N_u$) that respond to a stimulus value s, by firing at average rates $f_b(s)$, can similarly represent an entire set of functions. However, a function $h(s)$ that is not equal to any of the single neuron tuning curves must be represented by combining the responses of a number of units. This can be done using the network shown in figure 8.13. The average steady-state activity level of the output unit in this network, in response to stimulus value s, is given by equation 8.2,

function approximation

$$v(s) = \mathbf{w} \cdot \mathbf{u} = \mathbf{w} \cdot \mathbf{f}(s) = \sum_{b=1}^{N} w_b f_b(s) \,. \tag{8.51}$$

Note that we have replaced $\mathbf{u}$ with $\mathbf{f}(s)$ where $\mathbf{f}(s)$ is the vector with components $f_b(s)$. The network presented in chapter 7 that performs coordinate transformation is an example of this type of function approximator.

In equation 8.51, the input tuning curves $\mathbf{f}(s)$ act as a basis for representing the output function $h(s)$, and for this reason they are called basis functions. Different sets of basis functions can be used to represent a given set of output functions. A set of basis functions that can represent any member of a class of functions using a linear sum, as in equation 8.51, is called complete for this class. For the basis sets typically used in mathematics, such as the sines and cosines in a Fourier series, the weights in equation 8.51 are unique. When neural tuning curves are used to expand a function, the weights tend not to be unique, and the set of input functions is called overcomplete. In this chapter, we assume that the basis functions are held fixed, and only the weights are adjusted to improve output performance. It is also interesting to consider methods for learning the best basis functions for a particular application. One way of doing this is by applying an algorithm called backpropagation, which develops the basis functions guided by the output errors of the network. Other methods, which we consider in chapter 10, involve unsupervised learning.

basis functions

completeness

overcomplete

Suppose that the function-representation network of figure 8.13 is pro-

an integral in the limit $\Delta r \to 0$. In this limit, we can write

$$\lim_{\Delta r \to 0} \{H + \log_2 \Delta r\} = -\int dr\, p[r] \log_2 p[r]\,. \tag{4.17}$$

continuous entropy

Δr is best thought of as a limit on the resolution with which the firing rate can be measured. Unless this limit is known, the entropy of a probability density for a continuous variable can be determined only up to an additive constant. However, if two entropies computed with the same resolution are subtracted, the troublesome term involving Δr cancels, and we can proceed without knowing its precise value. All of the cases where we use equation 4.17 are of this form. The integral on the right side of equation 4.17 is sometimes called the differential entropy.

differential entropy

The noise entropy, for a continuous variable like the firing rate, can be written in a manner similar to the response entropy 4.17, except that the conditional probability density $p[r|s]$ is used:

continuous noise entropy

$$\lim_{\Delta r \to 0} \{H_{\text{noise}} + \log_2 \Delta r\} = -\int ds \int dr\, p[s] p[r|s] \log_2 p[r|s]\,. \tag{4.18}$$

The mutual information is the difference between the expressions in equations 4.17 and 4.18,

continuous mutual information

$$I_{\text{m}} = \int ds \int dr\, p[s] p[r|s] \log_2 \left(\frac{p[r|s]}{p[r]} \right). \tag{4.19}$$

Note that the factor of $\log_2 \Delta r$ cancels in the expression for the mutual information because both entropies are evaluated at the same resolution.

In chapter 3, we described the Fisher information as a local measure of how tightly the responses determine the stimulus. The Fisher information is local because it depends on the expected curvature of the likelihood $P[\mathbf{r}|s]$ (typically for the responses of many cells) evaluated at the true stimulus value. The mutual information is a global measure in the sense that it depends on the average overall uncertainty in the decoding distribution $p[s|\mathbf{r}]$, including values of s both close to and far from the true stimulus. If the decoding distribution $p[s|\mathbf{r}]$ has a single peak about the true stimulus, the Fisher information and the mutual information are closely related. In particular, for large numbers of neurons, the maximum likelihood estimator tends to have a sharply peaked Gaussian distribution, as discussed in chapter 3. In this case, the mutual information is, up to an additive constant, the logarithm of the Fisher information averaged over the distribution of stimuli.

4.2 Information and Entropy Maximization

Entropy and mutual information are useful quantities for characterizing the nature and efficiency of neural encoding and selectivity. Often, in addition to such characterizations, we seek to understand the computational

The threshold γ determines the dividing line between values of $\mathbf{w} \cdot \mathbf{u}$ that generate $+1$ and -1 outputs. The supervised learning task for the perceptron is to place each of N_S input patterns $\mathbf{u}^m$ into one of two classes designated by the desired binary output v^m. How well the perceptron performs this task depends on the nature of the classification. The weight vector and threshold define a subspace (a hyperplane) of dimension $N_u - 1$ (the subspace of points satisfying $\mathbf{w} \cdot \mathbf{u} = \gamma$, which is perpendicular to $\mathbf{w}$) that cuts the N_u-dimensional space of input vectors into two regions. It is possible for a perceptron to classify inputs perfectly only if a hyperplane exists that divides the input space into one half-space containing all the inputs corresponding to $v = +1$, and another half-space containing all those for $v = -1$. This condition is called linear separability. An instructive case to consider is when each component of each input vector and the associated output values are chosen randomly and independently, with equal probabilities of being $+1$ and -1. For large N_u, the maximum number of random associations that can be described by a perceptron for typical examples of this type is $2N_u$.

linear separability

For linearly separable inputs, a set of weights exists that allows the perceptron to perform perfectly. However, this does not mean that a Hebbian modification rule can construct such weights. A Hebbian rule based on equation 8.45 with $\alpha = N_u/N_S$ constructs the weight vector

$$\mathbf{w} = \frac{1}{N_u} \sum_{m=1}^{N_S} v^m \mathbf{u}^m \,. \tag{8.47}$$

To see how well such weights allow the perceptron to perform, we compute the output generated by one input vector, $\mathbf{u}^n$, chosen from the training set. For this example, we set $\gamma = 0$. Nonzero threshold values are considered later in the chapter.

With $\gamma = 0$, the value of v for input $\mathbf{u}^n$ is determined solely by the sign of $\mathbf{w} \cdot \mathbf{u}^n$. Using the weights of equation 8.47, we find

$$\mathbf{w} \cdot \mathbf{u}^n = \frac{1}{N_u} \left(v^n \mathbf{u}^n \cdot \mathbf{u}^n + \sum_{m \neq n} v^m \mathbf{u}^m \cdot \mathbf{u}^n \right) . \tag{8.48}$$

If we set $\sum_{m \neq n} v^m \mathbf{u}^m \cdot \mathbf{u}^n / N_u = \eta^n$ (where the superscript on η, as on v, is a label, not a power), then $v^n \mathbf{u}^n \cdot \mathbf{u}^n / N_u = v^n$ because $1^2 = (-1)^2 = 1$, so we can write

$$\mathbf{w} \cdot \mathbf{u}^n = v^n + \eta^n \,. \tag{8.49}$$

Substituting this expression into equation 8.46 to determine the output of the perceptron for the input $\mathbf{u}^n$, we see that the term η^n acts as a source of noise, interfering with the ability of the perceptron to generate the correct answer $v = v^n$.

We can think of η^n as a sample from a probability distribution over values of η. Consider the case for which all components of $\mathbf{u}^m$ and v^m for all m

ject to this constraint is a constant,

$$p[r] = \frac{1}{r_{\max}}\,, \tag{4.22}$$

independent of r. The entropy for this probability density, for finite firing-rate resolution Δr, is

$$H = \log_2 r_{\max} - \log_2 \Delta r = \log_2 \left(\frac{r_{\max}}{\Delta r}\right)\,. \tag{4.23}$$

histogram equalization

Equation 4.22 is the basis of a signal-processing technique called histogram equalization. Applied to neural responses, this is a procedure for tailoring the neuronal selectivity so that $p[r] = 1/r_{\max}$ in response to a set of stimuli over which the entropy is to be maximized. Suppose a neuron responds to a stimulus characterized by the parameter s by firing at a rate $r = f(s)$. For small Δs, the probability that the continuous stimulus variable falls in the range between s and $s + \Delta s$ is given in terms of the stimulus probability density by $p[s]\Delta s$. This produces a response that falls in the range between $f(s + \Delta s)$ and $f(s)$. If the response probability density takes its optimal value, $p[r] = 1/r_{\max}$, the probability that the response falls within this range is $|f(s + \Delta s) - f(s)|/r_{\max}$. Setting these two probabilities equal to each other, we find that $|f(s + \Delta s) - f(s)|/r_{\max} = p[s]\Delta s$.

Consider the case of a monotonically increasing response so that $f(s + \Delta s) > f(s)$ for positive Δs. Then, in the limit $\Delta s \to 0$, the equalization condition becomes

$$\frac{df}{ds} = r_{\max}\, p[s]\,, \tag{4.24}$$

which has the solution

$$f(s) = r_{\max} \int_{s_{\min}}^{s} ds'\, p[s']\,, \tag{4.25}$$

where $s_{\min}$ is the minimum value of s, which is assumed to generate no response. Thus, entropy maximization requires that the average firing rate of the responding neuron be proportional to the integral of the probability density of the stimulus.

Laughlin (1981) has provided evidence that responses of the large monopolar cell (LMC) in the visual system of the fly satisfy the entropy-maximizing condition. The LMC responds to contrast, and Laughlin measured the probability distribution of contrasts of natural scenes in habitats where the flies he studied live. The solid curve in figure 4.2 is the integral of this measured distribution. The data points in figure 4.2 are LMC responses as a function of contrast. These responses are measured as membrane potential fluctuation amplitudes, not as firing rates, but the analysis presented can be applied without modification. As figure 4.2 indicates, the response as a function of contrast is very close to the integrated probability density, suggesting that the LMC is using a maximum entropy encoding.

weakened input from neurons with $s_a > 0$. This asymmetrically broadens and shifts the tuning curve of the neuron with $s_a = 0$ to lower stimulus values. The leftward shift seen in figure 8.11A is a result of the temporal character of the plasticity rule and the temporal evolution of the stimulus during training. Note that the shift is in the direction opposite to the motion of the stimulus during training. This backward shift has an interesting interpretation. If the same time-dependent stimulus is presented again after training, the neuron with $s_a = 0$ will respond earlier than it did prior to training. Thus, the training experience causes neurons to develop responses that predict the behavior of the stimulus. Although we chose to discuss the neuron with $s_a = 0$ as a representative example, the responses of other neurons shift in a similar manner.

Asymmetric enlargements and backward shifts of neural response tuning curves similar to those predicted from temporally asymmetric LTP and LTD induction have been seen in recordings of hippocampal place cells in rats (see chapter 1) made by Mehta et al. (1997, 2000). Figure 8.11B shows the average location of place fields (the place-cell analog of receptive fields) recorded while a rat ran repeated laps around a closed track. Over time, the place field shifted backward along the track relative to the direction the rat moved.

8.4 Supervised Learning

In unsupervised learning, inputs are imposed during a training period and the output is determined by the network dynamics, using the current values of the weights. This means that the network and plasticity rule must uncover patterns and regularities in the input data (such as the direction of maximal variance) by themselves. In supervised learning, both a set of inputs and the corresponding desired outputs are imposed during training, so the network is essentially given the answer.

Two basic problems addressed in supervised learning are storage, which means learning the relationship between the input and output patterns provided during training, and generalization, which means being able to provide appropriate outputs for inputs that were not presented during training but are similar to those that were. The main tasks we consider within the context of supervised learning are classification of inputs into two categories and function approximation (or regression), in which the output of a network unit is trained to approximate a specified function of the input. Understanding generalization in such settings has been a major focus of theoretical investigations in statistics and computer science but lies outside the scope of our discussion.

Supervised Hebbian Learning

In supervised learning, paired input and output samples are presented during training. We label these pairs by $\mathbf{u}^m$ and v^m for $m = 1 \ldots N_S$, where

sider the entropy associated with individual neurons within the population. If $p[r_a] = \int \prod_{b \neq a} dr_b \, p[\mathbf{r}]$ is the probability density for response r_a from neuron a, its entropy is

$$H_a = -\int dr_a \, p[r_a] \log_2 p[r_a] - \log_2 \Delta r = -\int d\mathbf{r} \, p[\mathbf{r}] \log_2 p[r_a] - \log_2 \Delta r \, . \tag{4.27}$$

The true population entropy can never be greater than the sum of these individual neuron entropies over the entire population,

$$H \leq \sum_a H_a \, . \tag{4.28}$$

To prove this, we note that the difference between the full entropy and the sum of individual neuron entropies is

$$\sum_a H_a - H = \int d\mathbf{r} \, p[\mathbf{r}] \log_2 \left(\frac{p[\mathbf{r}]}{\prod_a p_a[r_a]} \right) \geq 0 \, . \tag{4.29}$$

The inequality follows from the fact that the middle expression is the KL divergence between the probability distributions $p[\mathbf{r}]$ and $\prod_a p[r_a]$, and a KL divergence is always nonnegative. Equality holds only if

$$p[\mathbf{r}] = \prod_a p[r_a] \, , \tag{4.30}$$

that is, if the responses of the neurons are statistically independent. Thus, the full response entropy is never greater than the sum of the entropies of the individual neurons in the population, and it reaches the limiting value when equation 4.30 is satisfied. A code that satisfies this condition is called a factorial code because the probability factorizes into a product of single neuron probabilities. When the population-response probability density factorizes, this implies that the individual neurons respond independently. The entropy difference in equation 4.29 has been suggested as a measure of redundancy.

factorial code

redundancy

Combining this result with the results of the previous section, we conclude that the maximum population-response entropy can be achieved by satisfying two conditions. First, the individual neurons must respond independently, which means that $p[\mathbf{r}] = \prod_a p[r_a]$ must factorize. Second, they must all have response probabilities that are optimal for whatever constraints are imposed (e.g., flat, exponential, or Gaussian). If the same constraint is imposed on every neuron, the second condition implies that every neuron must have the same response probability density. In other words, $p[r_a]$ must be the same for all a values, a property called probability equalization. This does not imply that all the neurons respond identically to every stimulus. Indeed, the conditional probabilities $p[r_a|s]$ must be different for different neurons if they are to act independently. We proceed by considering factorization and probability equalization as general principles of entropy maximization, without imposing explicit constraints.

factorization

probability equalization

in which the rows of the weight matrix $\mathbf{W}$ are different eigenvectors of the correlation matrix $\mathbf{Q}$, and all the elements of the recurrent weight matrix $\mathbf{M}$ are ultimately set to 0.

Goodall (1960) proposed an alternative scheme for decorrelating different output units. In his model, the feedforward weights $\mathbf{W}$ are kept constant, whereas the recurrent weights adapt according to the anti-Hebb rule *Goodall rule*

$$\tau_M \frac{d\mathbf{M}}{dt} = -(\mathbf{W} \cdot \mathbf{u})\mathbf{v} + \mathbf{I} - \mathbf{M}. \tag{8.41}$$

The minus sign in the term $-(\mathbf{W} \cdot \mathbf{u})\mathbf{v}$ embodies the anti-Hebbian modification. This term is nonlocal because the change in the weight of a given synapse depends on the total feedforward input to the postsynaptic neuron, not merely on the input at that particular synapse (recall that $\mathbf{v} \neq \mathbf{W} \cdot \mathbf{u}$ in this case because of the recurrent connections). The term $\mathbf{I} - \mathbf{M}$ prevents the weights from going to 0 by pushing them toward the identity matrix $\mathbf{I}$. Unlike 8.40, this rule requires the existence of autapses, synapses that a neuron makes onto itself (i.e., the diagonal elements of $\mathbf{M}$ are not 0).

If the Goodall plasticity rule converges and stops changing $\mathbf{M}$, the right side of equation 8.41 must vanish on average, which requires (using the definition of $\mathbf{K}$)

$$\langle (\mathbf{W} \cdot \mathbf{u})\mathbf{v} \rangle = \mathbf{I} - \mathbf{M} = \mathbf{K}^{-1}. \tag{8.42}$$

Multiplying both sides by $\mathbf{K}$ and using equation 8.30, we find

$$\langle \mathbf{v}\mathbf{v} \rangle = \langle (\mathbf{K} \cdot \mathbf{W} \cdot \mathbf{u})\mathbf{v} \rangle = \mathbf{I}. \tag{8.43}$$

This means that the outputs are decorrelated and also indicates histogram equalization in the sense, discussed in chapter 4, that all the elements of $\mathbf{v}$ have the same variance. Indeed, the Goodall algorithm can be used to implement the decorrelation and whitening discussed in chapter 4. Because the anti-Hebb and Goodall rules are based on linear models, they are capable of removing only second-order redundancy, meaning redundancy characterized by the covariance matrix. In chapter 10, we consider models that are based on eliminating higher orders of redundancy as well.

Timing-Based Plasticity and Prediction

Temporal Hebbian rules have been used in the context of multi-unit networks to store information about temporal sequences. To illustrate this, we consider a network with the architecture of figure 8.6. We study the effect of time-dependent synaptic plasticity, as given by equation 8.18, on the recurrent synapses of the model, leaving the feedforward synapses constant.

Suppose that before training the average response of output unit a to a stimulus characterized by a parameter s is given by the tuning curve $f_a(s)$,

viewing screen in terms of a single vector $\vec{x} = (x, y)$, or sometimes $\vec{y} = (x, y)$. Using this notation, the linear estimate of the response of a visual neuron discussed in chapter 2 can be written as

$$L(t) = \int_0^\infty d\tau \int d\vec{x}\, D(\vec{x}, \tau) s(\vec{x}, t - \tau)\,. \tag{4.32}$$

If the space-time receptive field $D(\vec{x}, \tau)$ is separable, $D(\vec{x}, \tau) = D_s(\vec{x}) D_t(\tau)$, and we can rewrite $L(t)$ as the product of integrals involving temporal and spatial filters. To keep the notation simple, we assume that the stimulus can also be separated, so that $s(\vec{x}, t) = s_s(\vec{x}) s_t(t)$. Then, $L(t) = L_s L_t(t)$ where

$$L_s = \int d\vec{x}\, D_s(\vec{x}) s_s(\vec{x}) \tag{4.33}$$

and

$$L_t(t) = \int_0^\infty d\tau\, D_t(\tau) s_t(t - \tau)\,. \tag{4.34}$$

In the following, we analyze the spatial and temporal components, D_s and D_t, separately by considering the information-carrying capacity of L_s and L_t. We study the spatial receptive fields of retinal ganglion cells in this section, and the temporal response properties of LGN cells in the next. Later, we discuss the application of information maximization ideas to primary visual cortex.

To derive appropriately optimal spatial filters, we consider an array of retinal ganglion cells with receptive fields covering a small patch of the retina. We assume that the statistics of the input are spatially (and temporally) stationary or translation-invariant. This means that all locations and directions in space (and all times), at least within the patch we consider, are equivalent. This equivalence allows us to give all of the receptive fields the same spatial structure, with the receptive fields of different cells merely being shifted to different points within the visual field. As a result, we write the spatial kernel describing a retinal ganglion cell with receptive field centered at the point $\vec{a}$ as $D_s(\vec{x} - \vec{a})$. The linear response of this cell is then

$$L_s(\vec{a}) = \int d\vec{x}\, D_s(\vec{x} - \vec{a}) s_s(\vec{x})\,. \tag{4.35}$$

Note that we are labeling the neurons by the locations $\vec{a}$ of the centers of their receptive fields rather than by an integer index such as i. This is a convenient labeling scheme that allows sums over neurons to be replaced by sums over parameters describing their receptive fields. The vectors $\vec{a}$ for the different neurons take on discrete values corresponding to the different neurons in the population. If many neurons are being considered, these discrete vectors may fill the range of receptive field locations quite densely. In this case, it is reasonable to approximate the large but discrete

rule

$$\tau_w \frac{dW_{ab}}{dt} = \langle v_a (u_b - W_{ab}) \rangle \,. \tag{8.37}$$

The elastic net modification rule is similar, but an additional term is included to make the maps smooth, because smoothing is not included in the rule that generates the activity in this case. The elastic net plasticity rule is

elastic net rule

$$\tau_w \frac{dW_{ab}}{dt} = \langle v_a (u_b - W_{ab}) \rangle + \beta \sum_{a' \text{ neighbor of } a} (W_{a'b} - W_{ab}) \,, \tag{8.38}$$

where the sum is over all points a' that are neighbors of a, and β is a parameter that controls the degree of smoothness in the map. The elastic net makes W_{ab} similar to $W_{a'b}$, if a and a' are nearby on the cortex, by reducing $(W_{a'b} - W_{ab})^2$. Both the feature-based and elastic net rules make $W_{ab} \to u_b$ when v_a is positive.

Figure 8.10A shows the results of an optical imaging experiment that reveals how ocularity and orientation selectivity are arranged across a region of the primary visual cortex of a macaque monkey. The dark lines show the boundaries of the ocular dominance stripes. The lighter lines show iso-orientation contours, which are locations where the preferred orientations are roughly the same. They indicate, by the regions they enclose, neighborhoods (called domains) of cells that favor similar orientations. They also show how these neighborhoods are arranged with respect to each other and to the ocular dominance stripes. There are singularities, called pinwheels, in the orientation map, where regions with different orientation preferences meet at a point. These tend to occur near the centers of the ocular dominance stripes. There are also linear zones where the iso-orientation domains are parallel. These tend to occur at, and run perpendicular to, the boundaries of the ocular dominance stripes.

Figure 8.10B shows the result of an elastic net model plotted in the same form as the macaque map of figure 8.10A. The similarity is evident and striking. Here $\mathbf{u} = (x, y, o, e\cos\theta, e\sin\theta)$ includes 5 stimulus features ($N_u = 5$): two (x, y) for retinal location, one (o) for ocularity, and two (θ, e) for the direction and strength of orientation. The self-organizing map can produce almost identical results, and noncompetitive and competitive Hebbian developmental algorithms can also lead to similar structures.

Anti-Hebbian Modification

We previously alluded to the problem of redundancy among multiple output neurons that can arise from feedforward Hebbian modification. The Oja rule of equation 8.16 for multiple output units, which takes the form

$$\tau_w \frac{dW_{ab}}{dt} = v_a u_b - \alpha v_a^2 W_{ab} \,, \tag{8.39}$$

$\tilde{Q}_{ss}$, which is real and nonnegative, is also called the stimulus power spectrum (see chapter 1). In terms of these Fourier transforms, equation 4.37 becomes

$$|\tilde{D}_s(\vec{\kappa})|^2 \tilde{Q}_{ss}(\vec{\kappa}) = \sigma_L^2 \,, \tag{4.41}$$

from which we find

$$|\tilde{D}_s(\vec{\kappa})| = \frac{\sigma_L}{\sqrt{\tilde{Q}_{ss}(\vec{\kappa})}} \,. \tag{4.42}$$

whitening filter

The linear kernel described by equation 4.42 exactly compensates for whatever dependence the Fourier transform of the stimulus correlation function has on the spatial frequency $\vec{\kappa}$, making the product $\tilde{Q}_{ss}(\vec{\kappa})|\tilde{D}_s(\vec{\kappa})|^2$ independent of $\vec{\kappa}$. This product is the power spectrum of L. The output of the optimal filter has a power spectrum that is independent of spatial frequency, and therefore has the same characteristics as white noise. Therefore, the kernel in equation 4.42 is called a whitening filter. Different spatial frequencies act independently in a linear system, so decorrelation and variance equalization require them to be utilized at equal signal strength.

The calculation we have performed determines only the amplitude $|\tilde{D}_s(\vec{\kappa})|$, and not $\tilde{D}_s(\vec{\kappa})$ itself. Thus, decorrelation and variance equalization do not uniquely specify the form of the linear kernel. We study some consequences of the freedom to choose different linear kernels satisfying equation 4.42 later in the chapter.

The spatial correlation function for natural scenes has been measured, with the result that $\tilde{Q}_{ss}(\vec{\kappa})$ is proportional to $1/|\vec{\kappa}|^2$ over the range it has been evaluated. The behavior near $\vec{\kappa} = 0$ is not well established, but the divergence of $1/|\vec{\kappa}|^2$ near $\vec{\kappa} = 0$ can be removed by setting $\tilde{Q}_{ss}(\vec{\kappa})$ proportional to $1/(|\vec{\kappa}|^2 + \kappa_0^2)$ where κ_0 is a constant. The stimuli of interest in the calculation of retinal ganglion receptive fields are natural images as they appear on the retina, not in the photographs from which the natural scenes statistics are measured. An additional factor must be included in $\tilde{Q}_{ss}(\vec{\kappa})$ to account for filtering introduced by the optics of the eye (the optical modulation transfer function). A simple model of the optical modulation transfer function results in an exponential correction to the stimulus correlation function,

optical modulation transfer function

$$\tilde{Q}_{ss}(\vec{\kappa}) \propto \frac{\exp(-\alpha|\vec{\kappa}|)}{|\vec{\kappa}|^2 + \kappa_0^2} \,, \tag{4.43}$$

with α a parameter. Substituting this into equation 4.42 gives the rather peculiar result that the amplitude $|\tilde{D}_s(\vec{\kappa})|$, being proportional to the inverse of the square root of $\tilde{Q}_{ss}$, is predicted to grow exponentially for large $|\vec{\kappa}|$. Whitening filters maximize entropy by equalizing the distribution of response power over the entire spatial frequency range. High spatial frequency components of images are relatively rare in natural scenes and, even if they occur, are greatly attenuated by the eye. The whitening filter compensates for this by boosting the responses to high spatial frequencies. Although this is the result of the entropy maximization calculation, it is not

of our simpler model). This example uses competitive Hebbian plasticity with nondynamic multiplicative weight normalization. Two weight matrices, $\mathbf{W}_R$ and $\mathbf{W}_L$, corresponding to right- and left-eye inputs, characterize the connectivity of the model. These are shown separately in figure 8.9A, which illustrates that the cortical cells develop retinotopically ordered receptive fields, and they segregate into alternating patches dominated by one eye or the other. The index a indicates the identity and cortical location of the ouput unit, whereas b indicates the retinal location of the center of the corresponding input unit. The ocular dominance pattern is easier to see in figure 8.9B, which shows the difference between the right- and left-eye weights, $\mathbf{W}_R - \mathbf{W}_L$, and figure 8.9C, which shows the net ocularity of the total input to each output neuron of the model ($\sum_b [W_R - W_L]_{ab}$ for each a). It is possible to analyze the structure shown in figure 8.9 and reveal the precise effect of the competition (i.e., the effect of changing the competition parameter δ in equation 8.34). Such an analysis shows, for example, that subtractive normalization of the synaptic weight is not necessary to ensure the robust development of ocular dominance, as it is in the noncompetitive case.

Feature-Based Models

Models of cortical map formation can get extremely complex when multiple neuronal selectivities, such as retinotopic location, ocular dominance, and orientation preference, are considered simultaneously. To deal with this, a class of more abstract models, called competitive feature-based models, has been developed. These use a general approach similar to the competitive Hebbian models discussed in the previous section. Feature-based models are not directly related to the biophysical reality of neuronal firing rates and synaptic strengths, but they provide a compact description of map development.

In a feedforward model, the selectivity of neuron a is determined by the feedforward weights W_{ab} for all values of b, describing how this neuron is connected to the input units of the network. The input units are driven by the stimulus and their responses reflect various stimulus features. Thus, selectivity in these models is determined by how the synaptic weights transfer the selectivities of the input units to the output units. The idea of a feature-based model is to simplify this by directly relating the output unit selectivities to the corresponding features of the stimulus. In feature-based models, the index b is not used to label different input units, but rather to label different features of the stimulus. N_u is thus equal to the number of parameters being used to characterize the stimulus. Also, the input variable u_b is set to the parameter used to characterize feature b of the stimulus. Similarly, W_{ab} does not describe the coupling between input unit b and output unit a, but instead it represents the selectivity of output unit a to stimulus feature b. For example, suppose $b = 1$ represents the location of a visual stimulus, and $b = 2$ represents its ocularity, the difference in strength between the left- and right-eye inputs. Then, u_1 and u_2 would be the location coordinate and the ocularity of the stimulus, and W_{a1} and W_{a2} would be the preferred stimulus location (the center of the

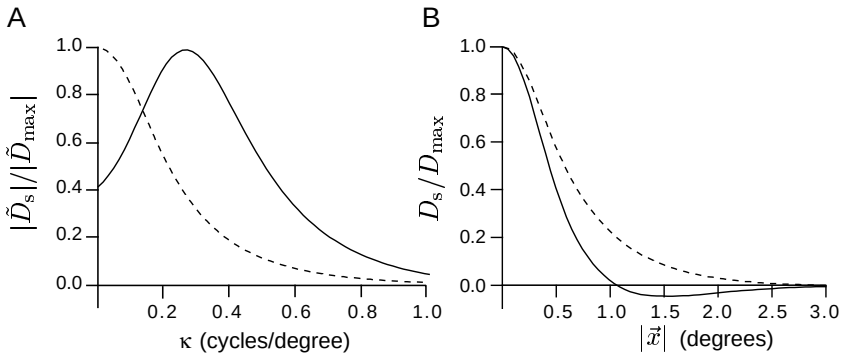

Figure 4.3 Receptive field properties predicted by entropy maximization and noise suppression of responses to natural images. (A) The amplitude of the predicted Fourier-transformed linear filters for low (solid curve) and high (dashed curve) input noise. $|\tilde{D}_s(\vec{\kappa})|$ is plotted relative to its maximum value. (B) The linear kernel as a function of the distance from the center of the receptive field for low (solid curve) and high (dashed curve) input noise. Observe the center-surround structure at low noise. $\tilde{D}_s(\vec{\kappa})$ is taken to be real, and $D_s(|\vec{x}|)$ is plotted relative to its maximum value. Parameter values used were $1/\alpha = 0.16$ cycles/degree, $k_0 = 0.16$ cycles/degree, and $\tilde{Q}_{\eta\eta}/\tilde{Q}_{ss}(0) = 0.05$ for the low-noise case and 1 for the high-noise case.

The calculation simplifies because we assume that the signal and noise terms are uncorrelated, so that $\langle s_s(\vec{x})\eta(\vec{y})\rangle = 0$. Then, the relevant cross-correlation for this problem is

$$\langle (s_s(\vec{x}) + \eta(\vec{x}))s_s(\vec{y})\rangle = Q_{ss}(\vec{x} - \vec{y})\,, \tag{4.44}$$

and the autocorrelation is

$$\langle (s_s(\vec{x}) + \eta(\vec{x}))(s_s(\vec{y}) + \eta(\vec{y}))\rangle = Q_{ss}(\vec{x} - \vec{y}) + Q_{\eta\eta}(\vec{x} - \vec{y})\,, \tag{4.45}$$

where Q_{ss} and $Q_{\eta\eta}$ are, respectively, the stimulus and noise autocorrelations functions. These results imply that the optimal noise filter is real and given, in terms of the Fourier transforms of Q_{ss} and $Q_{\eta\eta}$, by

noise filter

$$\tilde{D}_\eta(\vec{\kappa}) = \frac{\tilde{Q}_{ss}(\vec{\kappa})}{\tilde{Q}_{ss}(\vec{\kappa}) + \tilde{Q}_{\eta\eta}(\vec{\kappa})}\,. \tag{4.46}$$

Because the noise filter is designed so that its output matches the signal as closely as possible, we make the approximation of using the same whitening filter as before (equation 4.42). Combining the two, we find that

$$|\tilde{D}_s(\vec{\kappa})| \propto \frac{\sigma_L\sqrt{\tilde{Q}_{ss}(\vec{\kappa})}}{\tilde{Q}_{ss}(\vec{\kappa}) + \tilde{Q}_{\eta\eta}(\vec{\kappa})}\,. \tag{4.47}$$

Linear kernels resulting from equation 4.47, using equation 4.43 for the stimulus correlation function, are plotted in figure 4.3. For this figure, we have assumed that the input noise is white so that $\tilde{Q}_{\eta\eta}$ is independent of $\vec{\kappa}$. Both the amplitude of the Fourier transform of the kernel (figure 4.3A) and

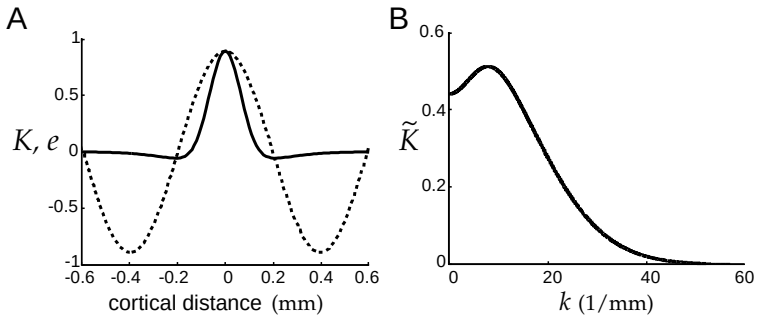

Figure 8.8 Hypothetical K function. (A) The solid line is $K(|a - a'|)$ given by the difference of two Gaussian functions. We have plotted this as a function of the distance between the cortical locations corresponding to the indices a and a', rather than of $|a - a'|$. The dotted line is the principal eigenvector plotted on the same scale. (B) $\tilde{K}$, the Fourier transform of K. This is also given by the difference of two Gaussians. As in A, we use cortical distance units and plot $\tilde{K}$ in terms of the spatial frequency k rather than the integer index μ.

the effects of excitation and inhibition by modeling competitive and cooperative aspects of how activity is generated in, and spread across, the cortex. The advantage of this approach is that it allows us to deal with complex, nonlinear features of Hebbian models in a more controlled and manageable way. As we have seen, competition between neurons is an important element in the development of realistic patterns of selectivity. Linear recurrent connections can produce only a limited amount of differentiation among network neurons, because they induce fairly weak competition between output units. As detailed in chapter 7, recurrent connections can lead to much stronger competition if the interactions are nonlinear. The model discussed in this section allows strongly nonlinear competition to arise in a Hebbian setting.

nonlinear competition

As mentioned in the previous paragraph, the model we discuss represents the effect of cortical processing in two somewhat abstract stages. One stage, modeling the effects of long-range inhibition, involves competition among all the cortical cells for feedforward input in a scheme related to that used in chapter 2 for contrast saturation. The second stage, modeling shorter-range excitation, involves cooperation in which neurons that receive feedforward input excite their neighbors.

In the first stage, the feedforward input for unit a, and that for all the other units, is fed through a nonlinear function to generate a competitive measure of the local excitation generated at location a,

$$z_a = \frac{\left(\sum_b W_{ab} u_b\right)^\delta}{\sum_{a'} \left(\sum_b W_{a'b} u_b\right)^\delta}. \tag{8.34}$$

The activities and weights are all assumed to be positive. The parameter δ controls the degree of competition. For large δ, only the largest feedforward input survives. The case $\delta = 1$ is quite similar to the linear recurrent connections of the previous section.

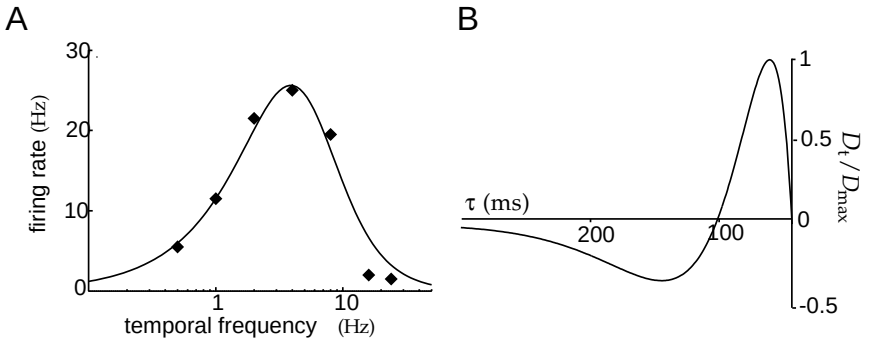

Figure 4.4 (A) Predicted (curve) and actual (diamonds) selectivity of an LGN cell as a function of temporal frequency. The predicted curve is based on the optimal linear filter $\tilde{D}_t(\omega)$ with $\omega_0 = 5.5$ Hz. (B) Causal, minimum phase, temporal form of the optimal filter. (Adapted from Dong and Atick, 1995; data in A from Saul and Humphrey, 1990.)

and $\tilde{D}_t(\omega)$ is given by an equation similar to 4.47,

$$|\tilde{D}_t(\omega)| \propto \frac{\sigma_L \sqrt{\tilde{Q}_{ss}(\omega)}}{\tilde{Q}_{ss}(\omega) + \tilde{Q}_{\eta\eta}(\omega)} . \tag{4.50}$$

In this case, $\tilde{Q}_{ss}(\omega)$ and $\tilde{Q}_{\eta\eta}(\omega)$ are the power spectra of the signal and the noise in the temporal domain.

Dong and Atick (1995) analyzed temporal receptive fields in the LGN in this way, under the assumption that a substantial fraction of the temporal redundancy of visual stimuli is removed in the LGN rather than in the retina. They determined that the temporal power spectrum of natural scenes has the form

$$\tilde{Q}_{ss}(\omega) \propto \frac{1}{\omega^2 + \omega_0^2} , \tag{4.51}$$

where ω_0 is a constant. The resulting filter, in both the temporal frequency and the time domains, is plotted in figure 4.4. Figure 4.4A shows the predicted and actual frequency responses of an LGN cell. This is similar to the plot in figure 4.3A, except that the result has been normalized to a realistic response level so that it can be compared with data. Because the optimization procedure determines only the amplitude of the Fourier transform of the linear kernel, $D_t(\tau)$ is not uniquely specified. To determine the temporal kernel, we require it to be causal ($D_t(\tau) = 0$ for $\tau < 0$) and impose a technical condition known as minimum phase, which assures that the output changes as rapidly as possible when the stimulus varies. Figure 4.4B shows the resulting form of the temporal filter. The space-time receptive fields shown in chapter 2 tend to change sign as a function of τ. The temporal filter in figure 4.4B has exactly this property.

An interesting test of the notion of optimal coding was carried out by Dan, Atick, and Reid (1996). They used both natural scene and white-noise stimuli while recording cat LGN cells. Figure 4.5A shows the power

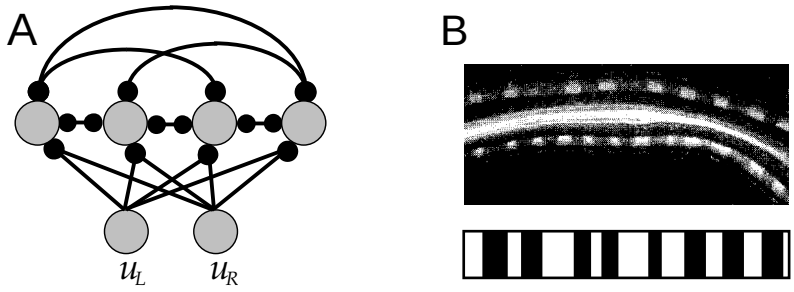

Figure 8.7 The development of ocular dominance in a Hebbian model. (A) The simplified model in which right- and left- eye inputs from a single retinal location drive an array of cortical neurons. (B) Ocular dominance maps. The upper panel shows an area of cat primary visual cortex radioactively labeled to distinguish regions activated by one eye or the other. The light and dark areas along the cortical regions at the top and bottom indicate alternating right- and left-eye innervation. The central region is white matter where fibers are not segregated by ocular dominance. The lower panel shows the pattern of innervation for a 512 unit model after Hebbian development. White and black regions denote units dominated by right- and left-eye projections respectively. (B data of S. LeVay, adapted from Nicholls et al., 1992.)

With fixed recurrent weights $\mathbf{M}$ and plastic feedforward weights $\mathbf{W}$, the effect of averaging Hebbian modifications over the training inputs is

$$\tau_w \frac{d\mathbf{W}}{dt} = \langle \mathbf{v}\mathbf{u} \rangle = \mathbf{K} \cdot \mathbf{W} \cdot \mathbf{Q}\,, \tag{8.31}$$

where $\mathbf{Q} = \langle \mathbf{u}\mathbf{u} \rangle$ is the input autocorrelation matrix. Equation 8.31 has the same form as the single unit equation 8.5, except that both $\mathbf{K}$ and $\mathbf{Q}$ affect the growth of $\mathbf{W}$.

We consider a highly simplified model of the development of ocular dominance maps that considers only a single direction across the cortex and a single point in the visual field. Figure 8.7A shows the simplified model, which has only two input activities, u_R and u_L, that have the correlation matrix of equation 8.25. These are connected to multiple output units through weight vectors $\mathbf{w}_\mathrm{R}$ and $\mathbf{w}_\mathrm{L}$. The output units are connected to each other through weights $\mathbf{M}$, so $\mathbf{v} = \mathbf{w}_\mathrm{R} u_\mathrm{R} + \mathbf{w}_\mathrm{L} u_\mathrm{L} + \mathbf{M} \cdot \mathbf{v}$. The index a denoting the identity of a given output unit also represents the location of that unit on the cortex. This linking of a to locations on the cortical surface allows us to interpret the results of the model in terms of a cortical map.

Writing $\mathbf{w}_+ = \mathbf{w}_\mathrm{R} + \mathbf{w}_\mathrm{L}$ and $\mathbf{w}_- = \mathbf{w}_\mathrm{R} - \mathbf{w}_\mathrm{L}$, the equivalent of equation 8.26 for these sum and difference vectors is

$$\tau_w \frac{d\mathbf{w}_+}{dt} = (q_\mathrm{S} + q_\mathrm{D})\mathbf{K} \cdot \mathbf{w}_+ \qquad \tau_w \frac{d\mathbf{w}_-}{dt} = (q_\mathrm{S} - q_\mathrm{D})\mathbf{K} \cdot \mathbf{w}_-\,. \tag{8.32}$$

As in the single-cell model of ocular dominance, subtractive normalization, which holds the value of $\mathbf{w}_+$ fixed while leaving the growth of $\mathbf{w}_-$ unaffected, eliminates the tendency for the cortical cells to become binocular. Then only the equation for $\mathbf{w}_-$ is relevant, and the growth of $\mathbf{w}_-$ is

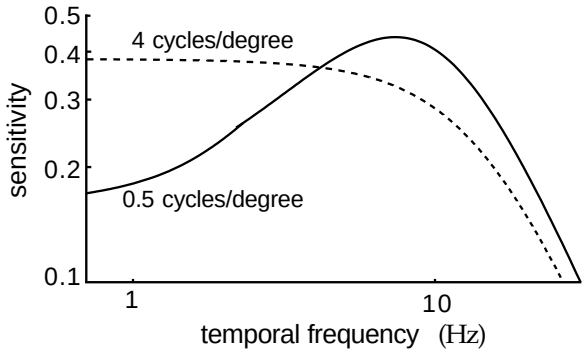

Figure 4.6 Dependence of temporal frequency tuning on preferred spatial frequency for space-time receptive fields derived from information maximization in the presence of noise. The curves show a transition from partial whitening in temporal frequency for low preferred spatial frequency (solid curve, 0.5 cycles/degree) to temporal summation for high preferred spatial frequency (dashed curve, 4 cycles/degree). (Adapted from Li, 1996.)

where $\alpha = 0.4$ cycle seconds/degree. This correlation function decreases both for high spatial and high temporal frequencies. Figure 4.6 shows how temporal selectivity for a combined noise and whitening filter, constructed using this stimulus power spectrum, changes for different preferred spatial frequencies. The basic idea is that components with fairly low stimulus power are boosted by the whitening filter, while those with very low stimulus power get suppressed by the noise filter. As shown by Li (1996), if a cell is selective for high spatial frequencies, the input signal rapidly falls below the noise (treated as white) as the temporal frequency of the input is increased. As a result, the noise filter of equation 4.46 causes the temporal response to be largest at 0 temporal frequency (dashed curve of figure 4.6). If instead the cell is selective for low spatial frequencies, the signal dominates the noise up to higher temporal frequencies, and the whitening filter causes the response to increase as a function of temporal frequency up to a maximum value where the noise filter begins to suppress the response (solid curve in figure 4.6). Receptive fields with preference for high spatial frequency thus act as low-pass temporal filters, and receptive fields with selectivity for low spatial frequency act as bandpass temporal filters.

Similar conclusions can be drawn concerning other joint selectivities. For example, color-selective (chrominance) cells tend to be selective for low temporal frequencies, because their input signal-to-noise ratio is lower than that for broadband (luminance) cells. There is also an interesting predicted relationship between ocular dominance and spatial frequency tuning due to the nature of the correlations between the two eyes. Optimal receptive fields with low spatial frequency tuning (for which the input signal-to-noise ratio is high) have enhanced sensitivity to differences between inputs coming from the two eyes. Receptive fields tuned to intermediate and high spatial frequencies suppress ocular differences.

activity across the synapse. This allows temporal Hebbian rules to exhibit a phenomenon called trace learning, where the term trace refers to the history of synaptic activity.

trace learning

We can approximate the final result of applying a temporal plasticity rule by integrating equation 8.18 from $t = 0$ to a large final time $t = T$, assuming that $w = 0$ initially, and shifting the integration variable to obtain

$$\mathbf{w} = \frac{1}{\tau_w} \int_0^T dt\, v(t) \int_{-\infty}^{\infty} d\tau\, H(\tau)\mathbf{u}(t-\tau)\,. \tag{8.27}$$

The approximation comes from ignoring both small contributions associated with the end points of the integral and the change in $\mathbf{v}$ produced during training by the modification of $\mathbf{w}$. Equation 8.27 shows that temporally dependent Hebbian plasticity depends on the correlation between the postsynaptic activity and the presynaptic activity temporally filtered by the function H.

Equation 8.27 (with a suitably chosen H) can be used to model the development of invariant responses. Neurons in inferotemporal cortex, for example, can respond selectively to particular objects independent of their location within a wide receptive field. The idea underlying the application of equation 8.27 is that objects persist in the visual environment for characteristic lengths of time as their images move across the retina. If the plasticity rule in equation 8.27 filters presynaptic activity over this persistence time, it strengthens synapses from all the presynaptic units that are activated by the image of the object while it persists and moves. As a result, the response of the postsynaptic cell comes to be independent of the position of the object, and position-invariant responses can be generated.

Multiple Postsynaptic Neurons

To study the effect of plasticity on multiple neurons, we introduce the network of figure 8.6, in which N_v output neurons receive input through N_u feedforward connections and from recurrent interconnections. A vector $\mathbf{v}$ represents the activities of the multiple output units, and the feedforward synaptic connections are described by a matrix $\mathbf{W}$, with the element W_{ab} giving the strength and sign of the synapse from input unit b to output unit a. The strength of the recurrent connection from output unit a' to output unit a is described by element $M_{aa'}$ of the recurrent weight matrix $\mathbf{M}$.

feedforward weight matrix $\mathbf{W}$

recurrent weight matrix $\mathbf{M}$

The activity vector for the output units of the network in figure 8.6 is determined by a linear version of the recurrent model of chapter 7,

$$\tau_r \frac{d\mathbf{v}}{dt} = -\mathbf{v} + \mathbf{W}\cdot\mathbf{u} + \mathbf{M}\cdot\mathbf{v}\,. \tag{8.28}$$

Provided that the real parts of the eigenvalues of $\mathbf{M}$ are less than 1, this equation has a stable fixed point with a steady-state output activity vector

satisfies

$$\dot{H} \leq -\langle r \rangle \int_0^\infty d\tau \, p[\tau] \log_2 (p[\tau] \Delta\tau) \, . \tag{4.52}$$

Poisson entropy rate

If a spike train is described by a homogeneous Poisson process with rate $\langle r \rangle$, we have $p[\tau] = \langle r \rangle \exp(-\langle r \rangle \tau)$, and the interspikes are statistically independent (chapter 1). Equation 4.52 is then an equality and, performing the integrals,

$$\dot{H} = \frac{\langle r \rangle}{\ln(2)} \left(1 - \ln(\langle r \rangle \Delta\tau)\right) \, . \tag{4.53}$$

We now turn to a more general calculation of the spike-train entropy. To make entropy calculations practical, a long spike train is broken into statistically independent subunits, and the total entropy is written as the sum of the entropies for the individual subunits. In the case of equation 4.52, the subunit was the interspike interval. If interspike intervals are not independent, and we wish to compute a result and not merely a bound, we must work with larger subunit descriptions. Strong et al. (1998) proposed a scheme that uses spike sequences of duration T_s as these basic subunits. Note that the variable T_s is used here to denote the duration of the spike sequence being considered, while T, which is much larger than T_s, is the duration of the entire spike train.

The time that a spike occurs is a continuous variable, so, as in the case of interspike intervals, a resolution must be specified when spike train entropies are computed. This can be done by dividing time into discrete bins of size Δt. We assume that the bins are small enough so that not more than one spike appears in a bin. Depending on whether or not a spike occurred within it, each bin is labeled by a 0 (no spike) or a 1 (spike). A spike sequence defined over a block of duration T_s is thus represented by a string of $T_s/\Delta t$ zeros and ones. We denote such a sequence by $B(t)$, where B is a $T_s/\Delta t$ bit binary number, and t specifies the time of the first bin in the sequence being considered. Both T_s and t are integer multiples of the bin size Δt.

The probability of a sequence B occurring at any time during the entire response is denoted by $P[B]$. This can be obtained by counting the number of times the sequence B occurs anywhere within the spike trains being analyzed (including overlapping cases). The spike-train entropy rate implied by this distribution is

$$\dot{H} = -\frac{1}{T_s} \sum_B P[B] \log_2 P[B] \, , \tag{4.54}$$

where the sum is over all the sequences B found in the data set, and we have divided by the duration T_s of a single sequence to obtain an entropy rate.

If the spike sequences in nonoverlapping intervals of duration T_s are independent, the full spike-train entropy rate is also given by equation 4.54.

and $w_- = w_R - w_L$ obey the uncoupled equations

$$\tau_w \frac{dw_+}{dt} = (q_S + q_D)w_+ \quad \text{and} \quad \tau_w \frac{dw_-}{dt} = (q_S - q_D)w_- . \tag{8.26}$$

Positive correlations between the two eyes are likely to exist ($q_D > 0$) after eye opening has occurred. This means that $q_S + q_D > q_S - q_D$, so, according to equations 8.26, w_+ grows more rapidly than w_-. Equivalently, $\mathbf{e}_1 = (1, 1)/\sqrt{2}$ is the principal eigenvector. The basic Hebbian rule thus predicts a final weight vector proportional to $\mathbf{e}_1$, which implies equal innervation from both eyes. This is not the observed outcome of cortical development.

Figure 8.3 shows, in a different example, that in the presence of constraints certain initial conditions can lead to final configurations that are not proportional to the principal eigenvector. However, in the present case initial conditions with $w_R > 0$ and $w_L > 0$ imply that $w_+ > w_-$, and this coupled with the faster growth of w_+ means that w_+ will always dominate over w_- at saturation. Multiplicative normalization does not change this situation.

To obtain ocular dominance in a Hebbian model, we must impose a constraint that prevents w_+ from dominating the final configuration. One way of doing this is to use equation 8.14, the Hebbian rule with subtractive normalization. This completely eliminates the growth of the weight vector in the direction of $\mathbf{e}_1$ (i.e., the increase of w_+) because, in this case, $\mathbf{e}_1$ is proportional to $\mathbf{n}$. On the other hand, it has no effect on growth in the direction $\mathbf{e}_2$ (i.e., the growth of w_-) because $\mathbf{e}_2 \cdot \mathbf{n} = 0$. Thus, with subtractive normalization, the weight vector grows parallel (or anti-parallel) to the direction $\mathbf{e}_2 = (1, -1)/\sqrt{2}$. The direction of this growth depends on initial conditions through the value of $\mathbf{w}(0) \cdot \mathbf{e}_2 = (w_R(0) - w_L(0))/\sqrt{2}$. If this is positive, w_R increases and w_L decreases; if it is negative, w_L increases and w_R decreases. Eventually, the decreasing weight will hit the saturation limit of 0, and the other weight will stop increasing due to the normalization constraint. At this point, total dominance by one eye or the other has been achieved. This simple model shows that ocular dominance can arise from Hebbian plasticity if there is sufficient competition between the growth of the left- and right-eye weights.

Hebbian Development of Orientation Selectivity

Hebbian models can also account for the development of the orientation selectivity displayed by neurons in primary visual cortex. The model of Hubel and Wiesel for generating an orientation-selective simple cell response by summing linear arrays of alternating ON and OFF LGN inputs was presented in chapter 2. The necessary pattern of LGN inputs can arise from Hebbian plasticity on the basis of correlations between the responses of different LGN cells and competition between ON and OFF units. Such a model can be constructed by considering a simple cell receiving input from ON-center and OFF-center cells of the LGN, and applying Hebbian

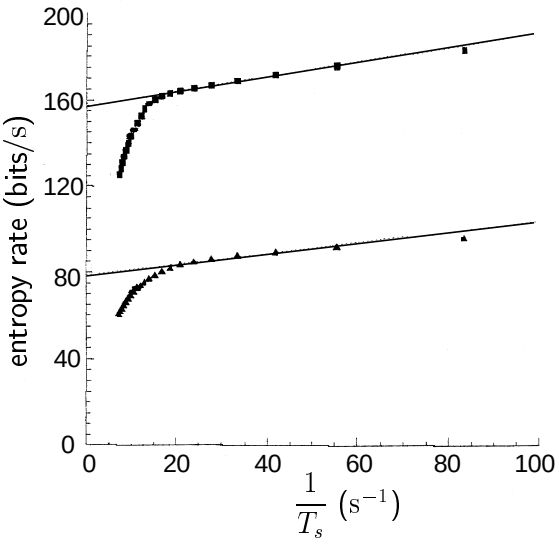

Figure 4.7 Entropy and noise entropy rates for the H1 visual neuron in the fly responding to a randomly moving visual image. The filled circles in the upper trace show the full spike-train entropy rate computed for different values of $1/T_s$. The straight line is a linear extrapolation to $1/T_s = 0$, which corresponds to $T_s \to \infty$. The lower trace shows the spike train noise entropy rate for different values of $1/T_s$. The straight line is again an extrapolation to $1/T_s = 0$. Both entropy rates increase as functions of $1/T_s$, and the true spike-train and noise entropy rates are overestimated at large values of $1/T_s$. At $1/T_s \approx 20/s$, there is a sudden shift in the dependence. This occurs when there is insufficient data to compute the spike sequence probabilities. The difference between the y intercepts of the two straight lines plotted is the mutual information rate. The resolution is $\Delta t = 3$ ms. (Adapted from Strong et al., 1998.)

stimuli in equation 4.6. The result is

$$\dot{H}_{\text{noise}} = -\frac{\Delta t}{T} \sum_t \left(\frac{1}{T_s} \sum_B P[B(t)] \log_2 P[B(t)] \right), \qquad (4.55)$$

where $T/\Delta t$ is the number of different t values being summed.

If equation 4.55 is based on finite-length spike sequences, it provides an upper bound on the noise entropy rate. The true noise entropy rate is estimated by performing a linear extrapolation in $1/T_s$ to $1/T_s = 0$, as was done for the spike-train entropy rate. The result, shown in figure 4.7, is a noise entropy of 79 bits/s for $\Delta t = 3$ ms. The information rate is obtained by taking the difference between the extrapolated values for the spike-train and noise entropy rates. The result for the fly H1 neuron used in figure 4.7 is an information rate of 157 - 79 = 78 bits/s or 1.8 bits/spike. Values in the range 1 to 3 bits/spike are typical results of such calculations for a variety of preparations.

Both the spike-train and noise entropy rates depend on Δt. The leading dependence, coming from the $\log_2 \Delta t$ term discussed previously, cancels

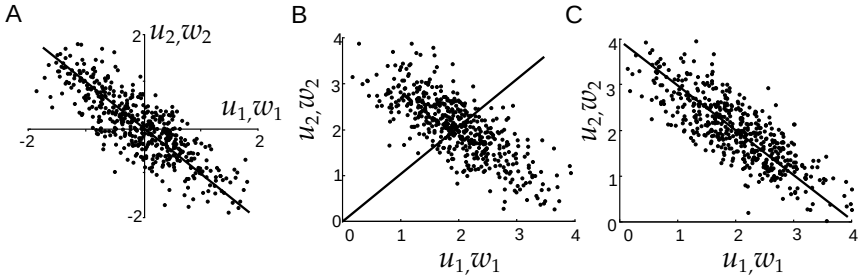

Figure 8.4 Unsupervised Hebbian learning and principal component analysis. The axes in these figures are used to represent the components of both $\mathbf{u}$ and $\mathbf{w}$. (A) The filled circles show the inputs $\mathbf{u} = (u_1, u_2)$ used during a training period while a Hebbian plasticity rule was applied. After training, the vector of synaptic weights was aligned parallel to the solid line. (B) Correlation-based modification with nonzero mean input. Input vectors were generated as in A except that the distribution was shifted to produce an average value $\langle \mathbf{u} \rangle = (2, 2)$. After a training period during which a Hebbian plasticity rule was applied, the synaptic weight vector was aligned parallel to the solid line. (C) Covariance-based modification. Points from the same distribution as in B were used while a covariance-based Hebbian rule was applied. The weight vector becomes aligned parallel to the solid line.

to the principal eigenvector of the correlation matrix of the inputs used during training. Figure 8.4A provides a geometric picture of the significance of this result. In this example, the basic Hebb rule was applied to a unit described by equation 8.2 with two inputs ($N_u = 2$). This model is not constrained to have positive $\mathbf{u}$ and v. The inputs used during the training period were chosen from a two-dimensional Gaussian distribution with unequal variances, resulting in the elliptical distribution of points seen in the figure. The initial weight vector $\mathbf{w}(0)$ was chosen randomly. The two-dimensional weight vector produced by a Hebbian rule is proportional to the principal eigenvector of the input correlation matrix. The line in figure 8.4A indicates the direction along which the final $\mathbf{w}$ lies, with the u_1 and u_2 axes used to represent w_1 and w_2 as well. The weight vector points in the direction along which the cloud of input points has the largest variance, a result with interesting implications.

Any unit that obeys equation 8.2 characterizes the state of its N_u inputs by a single number v, which is equal to the projection of $\mathbf{u}$ onto the weight vector $\mathbf{w}$. Intuition suggests, and a technique known as principal component analysis (PCA) formalizes, that setting the projection direction $\mathbf{w}$ proportional to the principal eigenvector $\mathbf{e}_1$ is often the optimal choice if a set of vectors is to be represented by, and reconstructed from, a set of single numbers through a linear relation. For example, if $\mathbf{e}_1$ is normalized so that $|\mathbf{e}_1| = 1$, the vectors $v\mathbf{w}$ with $v = \mathbf{w} \cdot \mathbf{u}$ and $\mathbf{w} = \mathbf{e}_1$ provide the best estimates that can be generated from single numbers (v) of the set of input vectors $\mathbf{u}$ used to construct $\mathbf{Q}$. Furthermore, the fact that projection along this direction maximizes the variance of the outputs that result can be interpreted using information theory. The entropy of a Gaussian distributed random variable with variance σ^2 grows with increasing variance as $\log_2 \sigma$. For Gaussian input statistics, and output that is corrupted

principal components analysis PCA

4.5 Appendix

Positivity of the Kullback-Leibler Divergence

Jensen's inequality

The logarithm is a concave function, which means that $\log_2\langle z\rangle \geq \langle \log_2 z\rangle$, where the angle brackets denote averaging with respect to some probability distribution and z is any positive quantity. The equality holds only if z is a constant. If we consider this relation, known as Jensen's inequality, with $z = Q[r]/P[r]$ and the average defined over the probability distribution $P[r]$, we find

$$-D_{\mathrm{KL}}(P,Q) = \sum_r P[r]\log_2\left(\frac{Q[r]}{P[r]}\right) \leq \log_2\left(\sum_r P[r]\frac{Q[r]}{P[r]}\right) = 0\,. \tag{4.56}$$

The last equality holds because $Q[r]$ is a probability distribution and thus satisfies $\sum_r Q[r] = 1$. Equation 4.56 implies that $D_{\mathrm{KL}}(P,Q) \geq 0$, with equality holding if and only if $P[r] = Q[r]$. A similar result holds for the Kullback-Leibler divergence between two probability densities,

$$D_{\mathrm{KL}}(p,q) = \int dr\, p[r]\log_2\left(\frac{p[r]}{q[r]}\right) \geq 0\,. \tag{4.57}$$

4.6 Annotated Bibliography

Information theory was created by Shannon (see **Shannon & Weaver, 1949**) largely as a way of understanding communication in the face of noise. **Cover & Thomas (1991)** provides a review, and **Rieke et al. (1997)** gives a treatment specialized to neural coding. Information theory and theories inspired by it, such as histogram equalization, were adopted in neuroscience and psychology as a way of understanding sensory transduction and coding, as discussed by **Barlow (1961)** and **Uttley (1979)**. We followed a more recent set of studies, inspired by Linkser (1988) and Barlow (1989), which have particularly focused on optimal coding in early vision; Atick & Redlich (1990), Plumbley (1991), Atick et al. (1992), **Atick (1992)**, van Hateren (1992; 1993), Li & Atick (1994a), Dong & Atick (1995), and Dan et al. (1996). Li & Atick (1994b) discuss the extension to joint selectivities of cells in V1; and Li & Atick (1994a) and Li (1996) treat stereo and motion sensitivities as examples.

The statistics of natural sensory inputs is reviewed by **Field (1987)**. Campbell & Gubisch (1966) estimated the optimal modulation transfer function.

We followed the technique of Strong et al. (1998) for computing the mutual information about a dynamical stimulus in spike trains. Bialek et al. (1993) presents an earlier approach based on stimulus reconstruction.

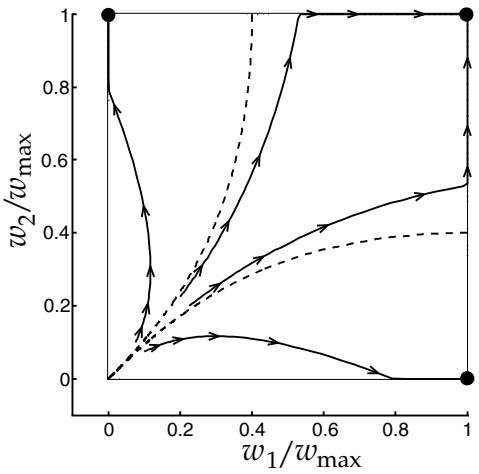

Figure 8.3 Hebbian weight dynamics with saturation. The correlation matrix of the input vectors had diagonal elements equal to 1 and off-diagonal elements of -0.4. The principal eigenvector, $\mathbf{e}_1 = (1, -1)/\sqrt{2}$, dominates the dynamics if the initial values of the weights are small enough (below or to the left of the dashed lines). This makes the weight vector move to the corners $(w_{\max}, 0)$ or $(0, w_{\max})$. However, starting the weights with larger values (between the dashed lines) allows saturation to occur at the corner $(w_{\max}, w_{\max})$. (Adapted from MacKay & Miller, 1990.)

Because the dot product corresponds to a projection of one vector onto another, Hebbian plasticity can be interpreted as producing an output proportional to the projection of the input vector onto the principal eigenvector of the correlation matrix of the inputs used during training. We discuss the significance of this result in the next section.

The proportionality sign in equation 8.21 hides the large multiplicative factor $\exp(\lambda_1 t/\tau_w)$ that is a consequence of the positive feedback inherent in Hebbian plasticity. One way to limit growth of the weight vector in equation 8.5 is to impose a saturation constraint. This can have significant effects on the outcome of Hebbian modification, including, in some cases, preventing the weight vector from ending up proportional to the principal eigenvector. Figure 8.3 shows examples of the Hebbian development of the weights in a case with just two inputs. For the correlation matrix used in this example, the principal eigenvector is $\mathbf{e}_1 = (1, -1)/\sqrt{2}$, so an analysis that ignored saturation would predict that one weight would increase while the other decreased. Which weight moves in which direction is controlled by the initial conditions. Given the constraints, this would suggest that $(w_{\max}, 0)$ and $(0, w_{\max})$ are the most likely final configurations. This analysis gives the correct answer only for the regions in figure 8.3 below or to the left of the dashed lines. Between the dashed lines, the final state is $\mathbf{w} = (w_{\max}, w_{\max})$ because the weights hit the saturation boundary before the exponential growth is large enough to allow the principal eigenvector to dominate.

Another way to eliminate the large exponential factor in the weights is to

postsynaptic activity during training. Among other things, this allows synaptic weights to store information about temporal sequences.

Rules in which synaptic plasticity is based on the relative timing of pre- and postsynaptic action potentials still require saturation constraints for stability, but, in spiking models, they can generate competition between synapses without further constraints or modifications. This is because different synapses compete to control the timing of postsynaptic spikes. Synapses that are able to evoke postsynaptic spikes rapidly get strengthened. These synapses then exert a more powerful influence on the timing of postsynaptic spikes, and they tend to generate spikes at times that lead to the weakening of other synapses that are less capable of controlling postsynaptic spike timing.

8.3 Unsupervised Learning

We now consider the computational properties of the different synaptic modification rules we have introduced in the context of unsupervised learning. Unsupervised learning provides a model for the effects of activity on developing neural circuits and the effects of experience on mature networks. We separate the discussion of unsupervised learning into cases involving a single postsynaptic neuron and cases in which there are multiple postsynaptic neurons.

A major area of research in unsupervised learning concerns the development of neuronal selectivity and the formation of cortical maps. Chapters 1 and 2 provided examples of the selectivities of neurons in various cortical areas to features of the stimuli to which they respond. In many cases, neuronal selectivities are arranged across the cortical surface in an orderly and regular pattern known as a cortical map. The patterns of connection that give rise to neuronal selectivities and cortical maps are established during development by both activity-independent and activity-dependent processes. A conventional view is that activity-independent mechanisms control the initial targeting of axons, determine the appropriate layer for them to innervate, and establish a coarse order or patterning in their projections. Other activity-independent and activity-dependent mechanisms then refine this order and help to create and preserve neuronal selectivities and cortical maps.

cortical map

Although the relative roles of activity-independent and activity-dependent processes in cortical development are the subject of extensive debate, developmental models based on activity-dependent plasticity have played an important role in suggesting key experiments and successfully predicting their outcomes. A detailed analysis of the more complex pattern-forming models that have been proposed is beyond the scope of this book. Instead, in this and later sections, we give a brief overview of some different approaches and the results that have been obtained. In this section, we consider the case of ocular dominance.

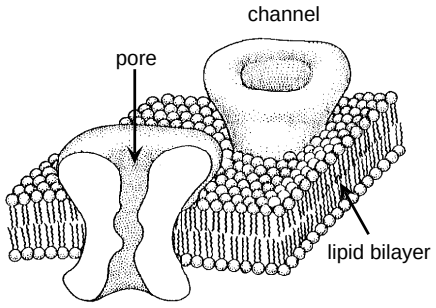

Figure 5.1 A schematic diagram of a section of the lipid bilayer that forms the cell membrane with two ion channels embedded in it. The membrane is 3 to 4 nm thick and the ion channels are about 10 nm long. (Adapted from Hille, 1992.)

ion channels

along its interior and exterior surfaces. Numerous ion-conducting channels embedded in the cell membrane (figure 5.1) lower the effective membrane resistance for ion flow to a value about 10,000 times smaller than that of a pure lipid bilayer. The resulting membrane conductance depends on the density and types of ion channels. A typical neuron may have a dozen or more different types of channels, anywhere from a few to hundreds of channels in a square micron of membrane, and hundreds of thousands to millions of channels in all. Many, but not all, channels are highly

channel selectivity

selective, allowing only a single type of ion to pass through them (to an accuracy of about 1 ion in 10^4). The capacity of channels for conducting ions across the cell membrane can be modified by many factors, including the membrane potential (voltage-dependent channels), the internal concentration of various intracellular messengers (Ca^{2+}-dependent channels, for example), and the extracellular concentration of neurotransmitters or neuromodulators (synaptic receptor channels, for example). The membrane

ion pumps

also contains selective pumps that expend energy to maintain differences in the concentrations of ions inside and outside the cell.

membrane potential

By convention, the potential of the extracellular fluid outside a neuron is defined to be 0. When a neuron is inactive, the excess internal negative charge causes the potential inside the cell membrane to be negative. This potential is an equilibrium point at which the flow of ions into the cell matches that out of the cell. The potential can change if the balance of ion flow is modified by the opening or closing of ion channels. Under normal conditions, neuronal membrane potentials vary over a range from about -90 to +50 mV. The order of magnitude of these potentials can be estimated from basic physical principles.

Membrane potentials are small enough to allow neurons to take advantage of thermal energy to help transport ions across the membrane, but are large enough so that thermal fluctuations do not swamp the signaling capabilities of the neuron. These conditions imply that potential differences across the cell membrane must lie in a range such that the energy gained or lost by an ion traversing the membrane is the same order of magnitude as its thermal energy. The thermal energy of an ion is about $k_B T$ where k_B

where α is a positive constant. This rule involves only information that is local to the synapse being modified (namely, the pre- and postsynaptic activities and the local synaptic weight), but its form is based more on theoretical arguments than on experimental data. The normalization it imposes is called multiplicative because the amount of modification induced by the second term in equation 8.16 is proportional to $\mathbf{w}$.

The stability of the Oja rule can be established by taking the dot product of equation 8.16 with the weight vector $\mathbf{w}$ to obtain

$$\tau_w \frac{d|\mathbf{w}|^2}{dt} = 2v^2(1 - \alpha|\mathbf{w}|^2) . \tag{8.17}$$

This indicates that $|\mathbf{w}|^2$ will relax over time to the value $1/\alpha$, which obviously prevents the weights from growing without bound, proving stability. It also induces competition between the different weights, because when one weight increases, the maintenance of a constant length for the weight vector forces other weights to decrease.

Timing-Based Rules

Experiments have shown that the relative timing of pre- and postsynaptic action potentials plays a critical role in determining the sign and amplitude of the changes in synaptic efficacy produced by activity. Figure 8.2 shows examples from an intracellular recording of a pair of cortical pyramidal cells in a slice experiment, and from an in vivo experiment on retinotectal synapses in a *Xenopus* tadpole. Both experiments involve repeated pairing of pre- and postsynaptic action potentials, and both show that the relative timing of these spikes is critical in determining the amount and type of synaptic modification that takes place. Synaptic plasticity occurs only if the difference in the pre- and postsynaptic spike times falls within a window of roughly ± 50 ms. Within this window, the sign of the synaptic modification depends on the order of stimulation. Presynaptic spikes that precede postsynaptic action potentials produce LTP. Presynaptic spikes that follow postsynaptic action potentials produce LTD. This is in accord with Hebb's original conjecture, because a synapse is strengthened only when a presynaptic action potential precedes a postsynaptic action potential and therefore can be interpreted as having contributed to it. When the order is reversed and the presynaptic action potential could not have contributed to the postsynaptic response, the synapse is weakened. The maximum amount of synaptic modification occurs when the paired spikes are separated by only a few milliseconds, and the evoked plasticity decreases to 0 as this separation increases.

Simulating the spike-timing dependence of synaptic plasticity requires a spiking model. However, an approximate model can be constructed on the basis of firing rates. The effect of pre- and postsynaptic timing can be included in a synaptic modification rule by including a temporal difference

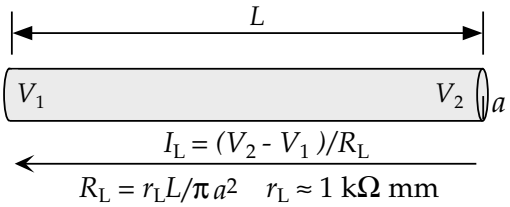

Figure 5.2 The longitudinal resistance of a cylindrical segment of neuronal cable with length L and radius a. The difference between the membrane potentials at the ends of this segment is related to the longitudinal current within the segment by Ohm's law, with R_L the longitudinal resistance of the segment. The arrow indicates the direction of positive current flow. The constant r_L is the intracellular resistivity, and a typical value is given.

single-channel conductance

its length and by r_L We approximate the channel pore as a tube of length 6 nm and opening area 0.15 nm^2. This gives an estimate of 0.15 nm^2 / (1 kΩ mm × 6 nm) ≈ 25 pS, which is the right order of magnitude for a channel conductance.

Membrane Capacitance and Resistance

electrotonic compactness

The intracellular resistance to current flow can cause substantial differences in the membrane potential measured in different parts of a neuron, especially during rapid transient excursions of the membrane potential from its resting value, such as action potentials. Neurons that have few of the long, narrow cable segments that produce high longitudinal resistances may have relatively uniform membrane potentials across their surfaces. Such neurons are termed electrotonically compact. For electrotonically compact neurons, or for less compact neurons in situations where spatial variations in the membrane potential are not thought to play an important functional role, the entire neuron may be adequately described by a single membrane potential. Here, we discuss the membrane capacitance and resistance using such a description. An analysis for the case of spatially varying membrane potentials is presented in chapter 6.

membrane capacitance C_m

specific membrane capacitance c_m

We have mentioned that there is typically an excess negative charge on the inside surface of the cell membrane of a neuron, and a balancing positive charge on its outside surface (figure 5.3). In this arrangement, the cell membrane creates a capacitance C_m, and the voltage across the membrane V and the amount of this excess charge Q are related by the standard equation for a capacitor, $Q = C_m V$. The membrane capacitance is proportional to the total amount of membrane or, equivalently, to the surface area of the cell. The constant of proportionality, called the specific membrane capacitance, is the capacitance per unit area of membrane, and it is approximately the same for all neurons, $c_m \approx 10$ nF/mm^2. The total capacitance C_m is the membrane surface area A times the specific capacitance, $C_m = c_m A$. Neuronal surface areas tend to be in the range 0.01 to 0.1 mm^2, so the membrane capacitance for a whole neuron is typically 0.1

growth by allowing the threshold to vary. The critical condition for stability is that θ_v must grow more rapidly than v as the output activity grows large. In one instantiation of the BCM rule with a sliding threshold, θ_v acts as a low-pass filtered version of v^2, as determined by the equation

sliding threshold

$$\tau_\theta \frac{d\theta_v}{dt} = v^2 - \theta_v \, . \tag{8.13}$$

Here τ_θ sets the time scale for modification of the threshold. This is usually slower than the presentation of individual presynaptic patterns, but faster than the rate at which the weights change, which is determined by τ_w. With a sliding threshold, the BCM rule implements competition between synapses because strengthening some synapses increases the postsynaptic firing rate, which raises the threshold and makes it more difficult for other synapses to be strengthened or even to remain at their current strengths.

Synaptic Normalization

The BCM rule stabilizes Hebbian plasticity by means of a sliding threshold that reduces synaptic weights if the postsynaptic neuron becomes too active. This amounts to using the postsynaptic activity as an indicator of the strengths of synaptic weights. A more direct way to stabilize a Hebbian plasticity rule is to add terms that depend explicitly on the weights. This typically leads to some form of weight normalization, which corresponds to the idea that postsynaptic neurons can support only a fixed total synaptic weight, so increases in some weights must be accompanied by decreases in others.

Normalization of synaptic weights involves imposing some sort of global constraint. Two types of constraints are typically used. If the synaptic weights are nonnegative, their growth can be limited by holding the sum of all the weights of the synapses onto a given postsynaptic neuron to a constant value. An alternative, which also works for weights that can be either positive or negative, is to constrain the sum of the squares of the weights instead of their linear sum. In either case, the constraint can be imposed either rigidly, requiring that it be satisfied at all times during the training process, or dynamically, requiring only that it be satisfied asymptotically at the end of training. We discuss one example of each type: a rigid scheme for imposing a constraint on the sum of synaptic weights, and a dynamic scheme for constraining the sum over their squares. Dynamic constraints can be applied in the former case and rigid constraints in the latter, but we restrict our discussion to two widely used schemes. We discuss synaptic normalization in connection with the basic Hebb rule, but the results we present can be applied to covariance rules as well. Weight normalization can drastically alter the outcome of a training procedure, and different normalization methods may lead to different outcomes.

membrane conductance

The membrane resistance is the inverse of the membrane conductance, and, like the capacitance, the conductance of a piece of cell membrane is proportional to its surface area. The constant of proportionality is the membrane conductance per unit area, but we write it as $1/r_m$, where r_m is called the specific membrane resistance. Conversely, the membrane resistance R_m is equal to r_m divided by the surface area. When a neuron is in a resting state, the specific membrane resistance is around 1 MΩ mm^2. This number is much more variable than the specific membrane capacitance. Membrane resistances vary considerably among cells, and under different conditions and at different times for a given neuron, depending on the number, type, and state of its ion channels. For total surface areas between 0.01 and 0.1 mm^2, the membrane resistance is typically in the range 10 to 100 MΩ. With a 100 MΩ membrane resistance, a constant current of 0.1 nA is required to hold the membrane potential 10 mV away from its resting value.

specific membrane resistance r_m

membrane time constant τ_m

The product of the membrane capacitance and the membrane resistance is a quantity with the units of time called the membrane time constant, $\tau_m = R_m C_m$. Because C_m and R_m have inverse dependences on the membrane surface area, the membrane time constant is independent of area and equal to the product of the specific membrane capacitance and resistance, $\tau_m = r_m c_m$. The membrane time constant sets the basic time scale for changes in the membrane potential and typically falls in the range between 10 and 100 ms.

Equilibrium and Reversal Potentials

Electric forces and diffusion are responsible for driving ions through channel pores. Voltage differences between the exterior and interior of the cell produce forces on ions. Negative membrane potentials attract positive ions into the neuron and repel negative ions. In addition, ions diffuse through channels because the ion concentrations differ inside and outside the neuron. These differences are maintained by the ion pumps within the cell membrane. The concentrations of Na^+ and Ca^{2+} are higher outside the cell than inside, so these ions are driven into the neuron by diffusion. K^+ is more concentrated inside the neuron than outside, so it tends to diffuse out of the cell.

equilibrium potential

It is convenient to characterize the current flow due to diffusion in terms of an equilibrium potential.This is defined as the membrane potential at which current flow due to electric forces cancels the diffusive flow. For channels that conduct a single type of ion, the equilibrium potential can be computed easily. The potential difference across the cell membrane biases the flow of ions into or out of a neuron. Consider, for example, a positively charged ion and a negative membrane potential. In this case, the membrane potential opposes the flow of ions out of the cell. Ions can cross the membrane and leave the interior of the cell only if their thermal energy suffices to overcome the energy barrier produced by the membrane poten-

$2\mathbf{w} \cdot d\mathbf{w}/dt$ and that $\mathbf{w} \cdot \mathbf{u} = v$, we find that $\tau_w d|\mathbf{w}|^2/dt = 2v^2$, which is always positive (except in the trivial case $v = 0$). Thus, the length of the weight vector grows continuously when the rule 8.3 is applied. To avoid unbounded growth, we must impose an upper saturation constraint. A lower limit is also required if the activity variables are allowed to be negative. Even with saturation, the basic Hebb rule fails to induce competition between different synapses.

Sometimes, synaptic modification is modeled as a discrete rather than continuous process, particularly if the learning procedure involves the sequential presentation of inputs. In this case, equation 8.5 is replaced by a discrete updating rule

$$\mathbf{w} \to \mathbf{w} + \epsilon \mathbf{Q} \cdot \mathbf{w} \tag{8.7}$$

where ϵ is a parameter, analogous to the learning rate $1/\tau_w$ in the continuous rule, that determines the amount of modification per application of the rule.

The Covariance Rule

If, as in Hebb's original conjecture, u and v are interpreted as representing firing rates (which must be positive), the basic Hebb rule describes only LTP. Experiments, such as the one shown in figure 8.1, indicate that synapses can depress in strength if presynaptic activity is accompanied by a low level of postsynaptic activity. High levels of postsynaptic activity, on the other hand, produce potentiation. These results can be modeled by a synaptic plasticity rule of the form

$$\tau_w \frac{d\mathbf{w}}{dt} = (v - \theta_v)\mathbf{u}\,, \tag{8.8}$$

where θ_v is a threshold that determines the level of postsynaptic activity above which LTD switches to LTP. As an alternative to equation 8.8, we can impose the threshold on the input rather than output activity, and write

postsynaptic threshold θ_v

$$\tau_w \frac{d\mathbf{w}}{dt} = v(\mathbf{u} - \boldsymbol{\theta}_u)\,. \tag{8.9}$$

Here, $\boldsymbol{\theta}_u$ is a vector of thresholds that determines the levels of presynaptic activities above which LTD switches to LTP. It is also possible to combine these two rules by subtracting thresholds from both the $\mathbf{u}$ and v terms, but this has the undesirable feature of predicting LTP when pre- and postsynaptic activity levels are both low.

presynaptic threshold $\boldsymbol{\theta}_u$

A convenient choice for the thresholds is the average value of the corresponding variable over the training period. In other words, we set the threshold in equation 8.8 to the average postsynaptic activity, $\theta_v = \langle v \rangle$, or the threshold vector in equation 8.9 to the average presynaptic activity vector, $\boldsymbol{\theta}_u = \langle \mathbf{u} \rangle$. As we did for equation 8.5, we use the relation $v = \mathbf{w} \cdot \mathbf{u}$ and

depolarization

hyperpolarization

shunting conductances

inhibitory and excitatory synapses

positive reversal potentials, they tend to depolarize a neuron (make its membrane potential less negative). K^+ conductances, with their negative E values, normally hyperpolarize a neuron (make its membrane potential more negative). Cl^- conductances, with reversal potentials near the resting potential, may pass little net current. Instead, their primary impact is to change the membrane resistance of the cell. Such conductances are sometimes called shunting, although all conductances "shunt", that is, increase the total conductance of a neuron. Synaptic conductances are also characterized by reversal potentials and are termed excitatory or inhibitory on this basis. Synapses with reversal potentials less than the threshold for action potential generation are typically called inhibitory, and those with reversal potentials above the action potential threshold are called excitatory.

The Membrane Current

The total current flowing across the membrane through all of its ion channels is called the membrane current of the neuron. By convention, the membrane current is defined as positive when positive ions leave the neuron and negative when positive ions enter the neuron. The total membrane current is determined by summing currents due to all of the different types of channels within the cell membrane, including voltage-dependent and synaptic channels. To facilitate comparisons between neurons of different sizes, it is convenient to use the membrane current per unit area of cell membrane, which we call i_m. The total membrane current is obtained from i_m by multiplying it by A, the total surface area of the cell.

membrane current per unit area i_m

We label the different types of channels in a cell membrane with an index i. As discussed in the last section, the current carried by a set of channels of type i with reversal potential E_i, vanishes when the membrane potential satisfies $V = E_i$. For many types of channels, the current increases or decreases approximately linearly when the membrane potential deviates from this value. The difference $V - E_i$ is called the driving force, and the membrane current per unit area due to the type i channels is written as $g_i(V - E_i)$. The factor g_i is the conductance per unit area, or specific conductance, due to these channels. Summing over the different types of channels, we obtain the total membrane current,

driving force

specific conductance g_i

membrane current

$$i_\mathrm{m} = \sum_i g_i(V - E_i)\,. \tag{5.5}$$

Sometimes a more complicated expression called the Goldman-Hodgkin-Katz formula is used to relate the membrane current to g_i and membrane potential (see Tuckwell, 1988; or Johnston and Wu, 1995), but we will restrict our discussion to the simpler relationship used in equation 5.5.

Much of the complexity and richness of neuronal dynamics arises because membrane conductances change over time. However, some of the factors that contribute to the total membrane current can be treated as relatively constant, and these are typically grouped together into a single term

In the models of plasticity that we study, the activity of each neuron is described by a continuous variable, not by a spike train. As in chapter 7, we use the letter u to denote the presynaptic level of activity and v to denote the postsynaptic activity. Normally, u and v represent the firing rates of the pre- and postsynaptic neurons, in which case they should be restricted to nonnegative values. Sometimes, to simplify the analysis, we ignore this constraint. An activity variable that takes both positive and negative values can be interpreted as the difference between a firing rate and a fixed background rate, or between the firing rates of two neurons being treated as a single unit. Finally, to avoid extraneous conversion factors in our equations, we take u and v to be dimensionless measures of the corresponding neuronal firing rates or activities. For example, u and v could be the firing rates of the pre- and postsynaptic neurons divided by their maximum or average values.

In the first part of this chapter, we consider unsupervised learning as applied to a single postsynaptic neuron driven by N_u presynaptic inputs with activities represented by u_b for $b = 1, 2, \ldots, N_u$, or collectively by the vector $\mathbf{u}$. In unsupervised learning, the postsynaptic activity v is evoked directly by the presynaptic activity $\mathbf{u}$. We describe v using a linear version of the firing-rate model discussed in chapter 7,

$$\tau_r \frac{dv}{dt} = -v + \mathbf{w} \cdot \mathbf{u} = -v + \sum_{b=1}^{N_u} w_b u_b \,, \tag{8.1}$$

where τ_r is a time constant that controls the firing-rate response dynamics. Recall that w_b is the synaptic weight that describes the strength of the synapse from presynaptic neuron b to the postsynaptic neuron, and $\mathbf{w}$ is the vector formed by all N_u synaptic weights. The individual synaptic weights can be either positive, representing excitation, or negative, representing inhibition. Equation 8.1 does not include any nonlinear dependence of the firing rate on the total synaptic input, not even rectification. Using such a linear firing-rate model considerably simplifies the analysis of synaptic plasticity. The restriction to nonnegative v either will be imposed by hand or, sometimes, will be ignored to simplify the analysis.

weight vector $\mathbf{w}$

The processes of synaptic plasticity are typically much slower than the dynamics characterized by equation 8.1. If, in addition, the stimuli are presented slowly enough to allow the network to attain its steady-state activity during training, we can replace the dynamic equation 8.1 by

$$v = \mathbf{w} \cdot \mathbf{u} \,, \tag{8.2}$$

which instantaneously sets v to the asymptotic steady-state value determined by equation 8.1. This is the equation we primarily use in our analysis of synaptic plasticity in unsupervised learning. Synaptic modification is included in the model by specifying how the vector $\mathbf{w}$ changes as a function of the pre- and postsynaptic levels of activity. The complex time course of plasticity seen in figure 8.1 is simplified by modeling only the longer-lasting changes.

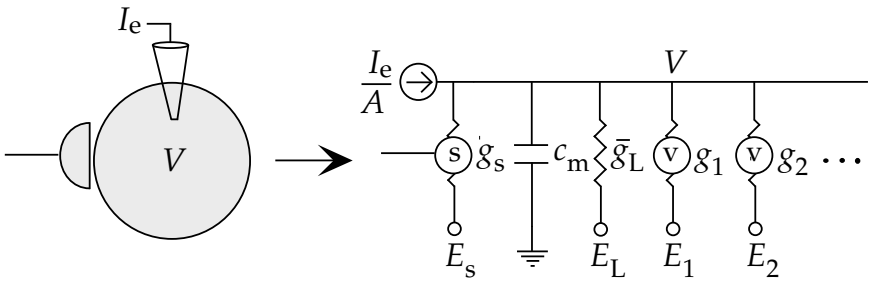

Figure 5.4 The equivalent circuit for a one-compartment neuron model. The neuron is represented, at the left, by a single compartment of surface area A with a synapse and a current-injecting electrode. At right is the equivalent circuit. The circled (s) indicates a synaptic conductance that depends on the activity of a presynaptic neuron. A single synaptic conductance g_s is indicated, but in general there may be several different types. The circled (v) indicates a voltage-dependent conductance, and I_e is the current passing through the electrode. The dots stand for possible additional membrane conductances.

this together, the basic equation for all single-compartment models is

$$c_m \frac{dV}{dt} = -i_m + \frac{I_e}{A}. \tag{5.6}$$

By convention, current that enters the neuron through an electrode is defined as positive-inward, whereas membrane current is defined as positive-outward. This explains the different signs for the currents in equation 5.6. The membrane current in equation 5.6 is determined by equation 5.5 and additional equations that specify the conductance variables g_i. The structure of such a model is the same as that of an electrical circuit, called the equivalent circuit, consisting of a capacitor and a set of variable and nonvariable resistors corresponding to the different membrane conductances. Figure 5.4 shows the equivalent circuit for a generic one-compartment model.

equivalent circuit

5.4 Integrate-and-Fire Models

A neuron will typically fire an action potential when its membrane potential reaches a threshold value of about -55 to -50 mV. During the action potential, the membrane potential follows a rapid, stereotyped trajectory and then returns to a value that is hyperpolarized relative to the threshold potential. As we will see, the mechanisms by which voltage-dependent K^+ and Na^+ conductances produce action potentials are well understood and can be modeled quite accurately. On the other hand, neuron models can be simplified and simulations can be accelerated dramatically if the biophysical mechanisms responsible for action potentials are not explicitly included in the model. Integrate-and-fire models do this by stipulating that an action potential occurs whenever the membrane potential of

Studies of plasticity and learning involve analyzing how synapses are affected by activity over the course of a training period. In this and the following chapters, we consider three types of training procedures. In unsupervised (also called self-supervised) learning, a network responds to a series of inputs during training solely on the basis of its intrinsic connections and dynamics. The network then self-organizes in a manner that depends on the synaptic plasticity rule being applied and on the nature of the inputs presented during training. We consider unsupervised learning in a more general setting called density estimation in chapter 10.

unsupervised learning

In supervised learning, which we consider in the last section of this chapter, a desired set of input-output relationships is imposed on the network by a "teacher" during training. Networks that perform particular tasks can be constructed in this way by letting a modification rule adjust their synapses until the desired computation emerges as a consequence of the training process. This is an alternative to explicitly specifying the synaptic weights, as was done in chapter 7. In this case, finding a biologically plausible teaching mechanism may not be a concern if the question being addressed is whether any weights can be found that allow a network to implement a particular function. In more biologically plausible examples of supervised learning, one network acts as the teacher for another network.

supervised learning

In chapter 9, we discuss a third form of learning, reinforcement learning, that is intermediate between these cases. In reinforcement learning, the network output is not constrained by a teacher, but evaluative feedback about network performance is provided in the form of reward or punishment. This can be used to control the synaptic modification process.

reinforcement learning

In this chapter we largely focus on activity-dependent synaptic plasticity of the Hebbian type, meaning plasticity based on correlations of pre- and postsynaptic firing. To ensure stability and to obtain interesting results, we often must augment Hebbian plasticity with more global forms of synaptic modification that, for example, scale the strengths of all the synapses onto a given neuron. These can have a major impact on the outcome of development or learning. Non-Hebbian forms of synaptic plasticity, such as those that modify synaptic strengths solely on the basis of pre- or postsynaptic firing, are likely to play important roles in homeostatic, developmental, and learning processes. Activity can also modify the intrinsic excitability and response properties of neurons. Models of such intrinsic plasticity show that neurons can be remarkably robust to external perturbations if they adjust their conductances to maintain specified functional characteristics. Intrinsic and synaptic plasticity can interact in interesting ways. For example, shifts in intrinsic excitability can compensate for changes in the level of input to a neuron caused by synaptic plasticity. It is likely that all of these forms of plasticity, and many others, are important elements of both the stability and the adaptability of nervous systems.

non-Hebbian plasticity

In this chapter, we describe and analyze basic correlation- and covariance-based synaptic plasticity rules in the context of unsupervised learning, and

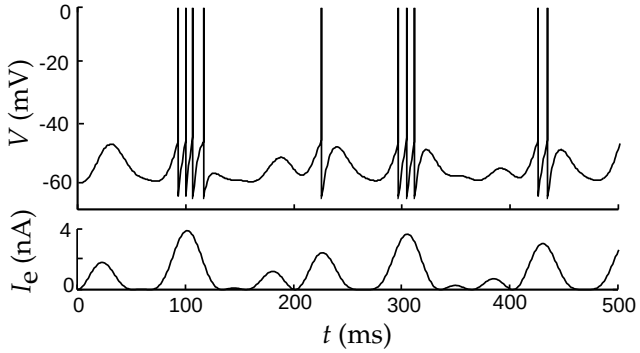

Figure 5.5 A passive integrate-and-fire model driven by a time-varying electrode current. The upper trace is the membrane potential, and the bottom trace the driving current. The action potentials in this figure are simply pasted onto the membrane potential trajectory whenever it reaches the threshold value. The parameters of the model are $E_L = V_{reset} = -65$ mV, $V_{th} = -50$ mV, $\tau_m = 10$ ms, and $R_m = 10$ MΩ.

time, the subthreshold potential $V(t)$ can easily be computed by solving equation 5.8, and is

$$V(t) = E_L + R_m I_e + (V(0) - E_L - R_m I_e) \exp(-t/\tau_m) , \tag{5.9}$$

where $V(0)$ is the value of V at time $t = 0$. This solution can be checked by substituting it into equation 5.8. It is valid for the integrate-and-fire model only as long as V stays below the threshold. Suppose that at $t = 0$, the neuron has just fired an action potential and is thus at the reset potential, so that $V(0) = V_{reset}$. The next action potential will occur when the membrane potential reaches the threshold, that is, at a time $t = t_{isi}$ when

$$V(t_{isi}) = V_{th} = E_L + R_m I_e + (V_{reset} - E_L - R_m I_e) \exp(-t_{isi}/\tau_m) . \tag{5.10}$$

By solving this for t_{isi}, the time of the next action potential, we can determine the interspike interval for constant I_e, or equivalently its inverse, which we call the interspike-interval firing rate of the neuron,

$$r_{isi} = \frac{1}{t_{isi}} = \left(\tau_m \ln \left(\frac{R_m I_e + E_L - V_{reset}}{R_m I_e + E_L - V_{th}} \right) \right)^{-1} . \tag{5.11}$$

This expression is valid if $R_m I_e > V_{th} - E_L$; otherwise $r_{isi} = 0$. For sufficiently large values of I_e, we can use the linear approximation of the logarithm ($\ln(1 + z) \approx z$ for small z) to show that

$$r_{isi} \approx \left[\frac{E_L - V_{th} + R_m I_e}{\tau_m (V_{th} - V_{reset})} \right]_+ , \tag{5.12}$$

which shows that the firing rate grows linearly with I_e for large I_e.

Figure 5.6A compares r_{isi} as a function of I_e, using appropriate parameter values, with data from current injection into a cortical neuron in vivo. The firing rate of the cortical neuron in figure 5.6A has been defined as the

8 Plasticity and Learning

8.1 Introduction

Activity-dependent synaptic plasticity is widely believed to be the basic phenomenon underlying learning and memory, and it is also thought to play a crucial role in the development of neural circuits. To understand the functional and behavioral significance of synaptic plasticity, we must study how experience and training modify synapses, and how these modifications change patterns of neuronal firing to affect behavior. Experimental work has revealed ways in which neuronal activity can affect synaptic strength, and experimentally inspired synaptic plasticity rules have been applied to a wide variety of tasks including auto- and heteroassociative memory, pattern recognition, storage and recall of temporal sequences, and function approximation.

Hebb rule

In 1949, Donald Hebb conjectured that if input from neuron A often contributes to the firing of neuron B, then the synapse from A to B should be strengthened. Hebb suggested that such synaptic modification could produce neuronal assemblies that reflect the relationships experienced during training. The Hebb rule forms the basis of much of the research done on the role of synaptic plasticity in learning and memory. For example, consider applying this rule to neurons that fire together during training due to an association between a stimulus and a response. These neurons would develop strong interconnections, and subsequent activation of some of them by the stimulus could produce the synaptic drive needed to activate the remaining neurons and generate the associated response. Hebb's original suggestion concerned increases in synaptic strength, but it has been generalized to include decreases in strength arising from the repeated failure of neuron A to be involved in the activation of neuron B. General forms of the Hebb rule state that synapses change in proportion to the correlation or covariance of the activities of the pre- and postsynaptic neurons.

Experimental work in a number of brain regions, including hippocampus, neocortex, and cerebellum, has revealed activity-dependent processes that can produce changes in the efficacies of synapses that persist for varying amounts of time. Figure 8.1 shows an example in which the data points

We model spike-rate adaptation by including an additional current in the model,

$$\tau_{\rm m}\frac{dV}{dt} = E_{\rm L} - V - r_{\rm m}g_{\rm sra}(V - E_K) + R_{\rm m}I_{\rm e}\,. \tag{5.13}$$

The spike-rate adaptation conductance $g_{\rm sra}$ has been modeled as a K^+ conductance so, when activated, it will hyperpolarize the neuron, slowing any spiking that may be occurring. We assume that this conductance relaxes to 0 exponentially with time constant $\tau_{\rm sra}$ through the equation

$$\tau_{\rm sra}\frac{dg_{\rm sra}}{dt} = -g_{\rm sra}\,. \tag{5.14}$$

Whenever the neuron fires a spike, $g_{\rm sra}$ is increased by an amount $\Delta g_{\rm sra}$, that is, $g_{\rm sra} \rightarrow g_{\rm sra} + \Delta g_{\rm sra}$. During repetitive firing, the current builds up in a sequence of steps causing the firing rate to adapt. Figures 5.6B and 5.6C compare the adapting firing pattern of a cortical neuron with the output of the model.

As discussed in chapter 1, the probability that a neuron fires is significantly reduced for a short period of time after the appearance of an action potential. Such a refractory effect is not included in the basic integrate-and-fire model. The simplest way of including an absolute refractory period in the model is to add a condition to the basic threshold crossing rule that forbids firing for a period of time immediately after a spike. Refractoriness can be incorporated in a more realistic way by adding a conductance similar to the spike-rate adaptation conductance discussed above, but with a faster recovery time and a larger conductance increment following an action potential. With a large increment, the current can essentially clamp the neuron to $E_{\rm K}$ following a spike, temporarily preventing further firing and producing an absolute refractory period. As this conductance relaxes back to 0, firing will be possible but initially less likely, producing a relative refractory period. When recovery is completed, normal firing can resume.

Another scheme that is sometimes used to model refractory effects is to raise the threshold for action-potential generation following a spike and then allow it to relax back to its normal value. Spike-rate adaptation can also be described by using an integrated version of the integrate-and-fire model known as the spike-response model, in which membrane potential waveforms are determined by summing precomputed postsynaptic potentials and after-spike hyperpolarizations. Finally, spike-rate adaptation and other effects can be incorporated into the integrate-and-fire framework by allowing the parameters $\overline{g}_{\rm L}$ and $E_{\rm L}$ in equation 5.7 to vary with time.

5.5 Voltage-Dependent Conductances

Most of the interesting electrical properties of neurons, including their ability to fire and propagate action potentials, arise from nonlinearities

III Adaptation and Learning

open probability P_i

given channel in the open state, and it is denoted by P_i. Thus, $g_i = \overline{g}_i P_i$. The dependence of a conductance on voltage, transmitter concentration, or other factors arises through effects on the open probability.

The open probability of a voltage-dependent conductance depends, as its name suggests, on the membrane potential of the neuron. In this chapter, we discuss models of two such conductances, the so-called delayed-rectifier K^+ and fast Na^+ conductances. The formalism we present, which is almost universally used to describe voltage-dependent conductances, was developed by Hodgkin and Huxley (1952) as part of their pioneering work showing how these conductances generate action potentials in the squid giant axon. Other conductances are modeled in chapter 6.

Persistent Conductances

Figure 5.8 shows cartoons of the mechanisms by which voltage-dependent channels open and close as a function of membrane potential. Channels are depicted for two different types of conductances, termed persistent (figure 5.8A) and transient (figure 5.8B). We begin by discussing persistent conductances. Figure 5.8A shows a swinging gate attached to a voltage sensor that can open or close the pore of the channel. In reality, channel gating mechanisms involve complex changes in the conformational structure of the channel, but the simple swinging gate picture is sufficient if we are interested only in the current-carrying capacity of the channel. A channel that acts as if it had a single type of gate (although, as we will see, this is actually modeled as a number of identical subgates), like the channel in figure 5.8A, produces what is called a persistent or noninactivating conductance. Opening of the gate is called activation of the conductance, and gate closing is called deactivation. For this type of channel, the probability that the gate is open, P_K, increases when the neuron is depolarized and decreases when it is hyperpolarized. The delayed-rectifier K^+ conductance that is responsible for repolarizing a neuron after an action potential is such a persistent conductance.

channel gate

activation
deactivation

The opening of the gate that describes a persistent conductance may involve a number of conformational changes. For example, the delayed-rectifier K^+ conductance is constructed from four identical subunits, and it appears that all four must undergo a structural change for the channel to open. In general, if k independent, identical events are required for a channel to open, P_K can be written as

$$P_K = n^k \,, \tag{5.15}$$

where n is the probability that any one of the k independent gating events has occurred. Here, n, which varies between 0 and 1, is called a gating or an activation variable, and a description of its voltage and time dependence amounts to a description of the conductance. We can think of n as the probability of an individual subunit gate being open, and $1 - n$ as the probability that it is closed.

activation variable n

where k is a constant, as can be verified by differentiating the right side. The nonconstant part of the right side of this equation is just (minus) the entropy associated with the binary variable v_a. In fact,

$$\sum_{a=1}^{N_v} \int_0^{I_a} dz_a \, z_a F'(z_a) = \langle \ln Q[\mathbf{v}] \rangle_Q + N_v k \,, \tag{7.62}$$

where the average is over all values of $\mathbf{v}$ weighted by their probabilities $Q[\mathbf{v}]$.

To evaluate the remaining terms in equation 7.60, we note that because the components of $\mathbf{v}$ are binary and independent for the Boltzmann machine, relations such as $\langle v_a \rangle_Q = F(I_a)$ and $\langle v_a v_b \rangle_Q = F(I_a)F(I_b)$ (for $a \neq b$) are valid. Then, using equation 7.56, we find

$$\sum_{a=1}^{N_v} \left(-h_a F(I_a) - \frac{1}{2} \sum_{a'=1}^{N_v} F(I_a) M_{aa'} F(I_{a'}) \right) = \langle E(\mathbf{v}) \rangle_Q \,. \tag{7.63}$$

Similarly, from equation 7.57, we can show that

$$\langle \ln P[\mathbf{v}] \rangle_Q = \langle -E(\mathbf{v}) \rangle_Q - \ln Z \,. \tag{7.64}$$

Combining the results of equations 7.62, 7.63, and 7.64, we obtain

$$L(\mathbf{I}) = \langle \ln Q[\mathbf{v}] - \ln P[\mathbf{v}] \rangle_Q + N_v k - \ln Z \,. \tag{7.65}$$

which gives equation 7.59 with $K = N_v k - \log Z$ because $\langle \ln Q[\mathbf{v}] - \ln P[\mathbf{v}] \rangle_Q$ is, by definition, the Kullback-Leibler divergence $D_{\mathrm{KL}}(Q, P)$. Note that in this (and subsequent) chapters, we define the Kullback-Leibler divergence using a natural logarithm, rather than the base 2 logarithm used in chapter 4. The two definitions differ only by an overall multiplicative constant.

7.9 Annotated Bibliography

Wilson & Cowan (1972, 1973) provides pioneering analyses of firing-rate models. Subsequent treatments related to the discussion in this chapter are presented in **Abbott (1994)**, **Ermentrout (1998)**, **Gerstner (1998)**, Amit & Tsodyks (1991a, 1991b), and Bressloff & Coombes (2000). The notion of a regular repeating unit of cortical computation dates back to the earliest investigations of cortex and is discussed by Douglas & Martin (1998).

Our discussion of the feedforward coordinate transformation model is based on Pouget & Sejnowski (1995, 1997) and Salinas & Abbott (1995), which built on theoretical work by Zipser & Andersen (1988) concerning parietal gain fields (see Andersen, 1989). Amplification by recurrent circuits is discussed in Douglas et al. (1995) and **Abbott (1994)**. We followed

The first term in equation 5.16 describes the opening process, and the second term the closing process (hence the minus sign) that lowers the probability of being in the configuration with an open subunit gate. Equation 5.16 can be written in another useful form by dividing through by $\alpha_n(V) + \beta_n(V)$,

gating equation

$$\tau_n(V)\frac{dn}{dt} = n_\infty(V) - n\,, \tag{5.17}$$

where

$\tau_n(V)$

$$\tau_n(V) = \frac{1}{\alpha_n(V) + \beta_n(V)} \tag{5.18}$$

and

$n_\infty(V)$

$$n_\infty(V) = \frac{\alpha_n(V)}{\alpha_n(V) + \beta_n(V)}. \tag{5.19}$$

Equation 5.17 indicates that for a fixed voltage V, n approaches the limiting value $n_\infty(V)$ exponentially with time constant $\tau_n(V)$.

The key elements in the equation that determines n are the opening and closing rate functions $\alpha_n(V)$ and $\beta_n(V)$. These are obtained by fitting experimental data. It is useful to discuss the form that we expect these rate functions to take on the basis of thermodynamic arguments. The state transitions described by α_n, for example, are likely to be rate-limited by barriers requiring thermal energy. These transitions involve the movement of charged components of the gate across part of the membrane, so the height of these energy barriers should be affected by the membrane potential. The transition requires the movement of an effective charge, which we denote by qB_α, through the potential V. This requires an energy $qB_\alpha V$. The constant B_α reflects both the amount of charge being moved and the distance over which it travels. The probability that thermal fluctuations will provide enough energy to surmount this energy barrier is proportional to the Boltzmann factor, $\exp(-qB_\alpha V/k_\mathrm{B}T)$. Based on this argument, we expect α_n to be of the form

$$\alpha_n(V) = A_\alpha \exp(-qB_\alpha V/k_\mathrm{B}T) = A_\alpha \exp(-B_\alpha V/V_T) \tag{5.20}$$

for some constant A_α. The closing rate β_n should be expressed similarly, except with different constants A_β and B_β. From equation 5.19, we then find that $n_\infty(V)$ is expected to be a sigmoidal function

$$n_\infty(V) = \frac{1}{1 + (A_\beta/A_\alpha)\exp((B_\alpha - B_\beta)V/V_T)}\,. \tag{5.21}$$

For a voltage-activated conductance, depolarization causes n to grow toward 1, and hyperpolarization causes n to shrink toward 0. Thus, we expect that the opening rate, α_n, should be an increasing function of V (and thus $B_\alpha < 0$), and β_n should be a decreasing function of V (and thus $B_\beta > 0$). Examples of the functions we have discussed are plotted in figure 5.9.

can be constructed on the basis of the deterministic synaptic current dynamics of a firing-rate model. In this case, $\mathbf{I}$ is determined by the dynamic equation 7.39 rather than by equation 7.54, with the function F in equation 7.39 set to the same sigmoidal function as in equation 7.55. The output v_a is determined from I_a at discrete times (integer multiples of Δt). The rule used for this is not the deterministic relationship $v_a = F(I_a)$ used in the firing-rate version of the model. Instead, v_a is determined from I_a stochastically, being set to either 1 or 0 with probability $F(I_a)$ or $1 - F(I_a)$ respectively. Thus, although the mean-field formulation for $\mathbf{I}$ is deterministic, $\mathbf{I}$ is used to generate a probability distribution over a binary output vector $\mathbf{v}$. Because $v_a = 1$ has probability $F(I_a)$ and $v_a = 0$ has probability $1 - F(I_a)$, and the units are independent, the probability distribution for the entire vector $\mathbf{v}$ is

$$Q[\mathbf{v}] = \prod_{a=1}^{N_v} F(I_a)^{v_a} (1 - F(I_a))^{1-v_a} . \tag{7.58}$$

This is called the mean-field distribution for the Boltzmann machine. Note that this distribution (and indeed $\mathbf{v}$ itself) plays no role in the dynamics of the mean-field formulation of the Boltzmann machine. It is, rather, a way of interpreting the outputs.

mean-field distribution

We have presented two formulations of the Boltzmann machine, Gibbs sampling and the mean-field approach, that lead to the two distributions $P[\mathbf{v}]$ and $Q[\mathbf{v}]$ (equations 7.57 and 7.58). The Lyapunov function of equation 7.40, which decreases steadily under the dynamics of equation 7.39 until a fixed point is reached, provides a key insight into the relationship between these two distributions. In the appendix to this chapter, we show that this Lyapunov function can be expressed as

$$L(\mathbf{I}) = D_{\mathrm{KL}}(Q, P) + K , \tag{7.59}$$

where K is a constant, and D_{KL} is the Kullback-Leibler divergence (see chapter 4). $D_{\mathrm{KL}}(Q, P)$ is a measure of how different the two distributions Q and P are from each other. The fact that the dynamics of equation 7.39 reduces the Lyapunov function to a minimum value means that it also reduces the difference between Q and P, as measured by the Kullback-Leibler divergence. This offers an interesting interpretation of the mean-field dynamics; it modifies the current value of the vector $\mathbf{I}$ until the distribution of binary output values generated by the mean-field formulation of the Boltzmann machine matches as closely as possible (finding at least a local minimum of $D_{\mathrm{KL}}(Q, P)$) the distribution generated by Gibbs sampling. In this way, the mean-field procedure can be viewed as an approximation of Gibbs sampling.

The power of the Boltzmann machine lies in the relationship between the distribution of output values, equation 7.57, and the quadratic energy function of equation 7.56. This makes it possible to determine how changing the weights $\mathbf{M}$ affects the distribution of output states. In chapter 8, we present a learning rule for the weights of the Boltzmann machine that allows $P[\mathbf{v}]$ to approximate a probability distribution extracted from a set

activation variable m

is open is written as m^k, where m is an activation variable similar to n and k is an integer. Hodgkin and Huxley used $k = 3$ for their model of the fast Na^+ conductance. The ball in figure 5.8B acts as the second gate. The probability that the ball does not block the channel pore is written as h and is

inactivation variable h

called the inactivation variable. The activation and inactivation variables m and h are distinguished by having opposite voltage dependences. Depolarization causes m to increase and h to decrease, and hyperpolarization decreases m while increasing h.

For the channel in figure 5.8B to conduct, both gates must be open, and assuming the two gates act independently, this has probability

$$P_{\mathrm{Na}} = m^k h\,. \tag{5.23}$$

This is the general form used to describe the open probability for a transient conductance. We could raise the h factor in this expression to an arbitrary power, as we did for m, but we omit this complication to streamline the discussion. The activation m and inactivation h, like all gating variables, vary between 0 and 1. They are described by equations identical to 5.16, except that the rate functions α_n and β_n are replaced by either α_m and β_m, or α_h and β_h. These rate functions were fitted by Hodgkin and Huxley using the equations (in units of 1/ms with V in mV)

$$\begin{aligned} \alpha_m &= \frac{.1(V+40)}{1-\exp(-.1(V+40))} \qquad & \beta_m &= 4\exp(-.0556(V+65)) \\ \alpha_h &= .07\exp(-.05(V+65)) & \beta_h &= 1/(1+\exp(-.1(V+35)))\,. \end{aligned} \tag{5.24}$$

Functions $m_\infty(V)$ and $h_\infty(V)$ describing the steady-state activation and inactivation levels, and voltage-dependent time constants for m and h can be defined as in equations 5.19 and 5.18. These are plotted in figure 5.10. For comparison, $n_\infty(V)$ and $\tau_n(V)$ for the K^+ conductance are also plotted. Note that $h_\infty(V)$, because it corresponds to an inactivation variable, is flipped relative to $m_\infty(V)$ and $n_\infty(V)$, so that it approaches 1 at hyperpolarized voltages and 0 at depolarized voltages.

The presence of two factors in equation (5.23) gives a transient conductance some interesting properties. To turn on a transient conductance maximally, it may first be necessary to hyperpolarize the neuron below its resting potential and then to depolarize it. Hyperpolarization raises

deinactivation

the value of the inactivation h, a process called deinactivation. The second step, depolarization, increases the value of m, which is activation. Only when m and h are both nonzero is the conductance turned on. Note that the conductance can be reduced in magnitude by either decreasing m or

inactivation

h. Decreasing h is called inactivation to distinguish it from decreasing m, which is deactivation.

Hyperpolarization-Activated Conductances

Persistent currents act as if they are controlled by an activation gate, while transient currents act as if they have both an activation and an inactivation

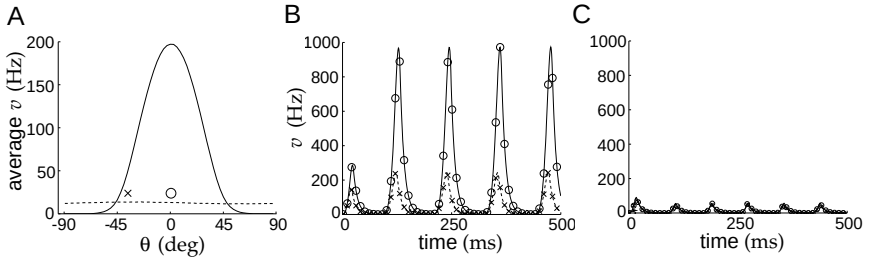

Figure 7.23 Selective amplification in an excitatory-inhibitory network. (A) Time-averaged response of the network to a tuned input with $\Theta = 0^\circ$ (solid curve) and to an untuned input (dashed curve). Symbols "o" and "x" mark the 0° and -37° points seen in B and C. (B) Activities over time of neurons with preferred angles of $\theta = 0^\circ$ (solid curve) and $\theta = -37^\circ$ (dashed curve) in response to a modulated input with $\Theta = 0^\circ$. (C) Activities of the same units shown in B to a constant input. The lines lie on top of each other, showing that the two units respond identically. The parameters are $\tau_E = \tau_I = 10$ ms, $h_I = 0$, $M_{EI} = -\delta(\theta - \theta')/\rho_\theta$, $M_{EE} = (1/\pi\rho_\theta)[5.9 + 7.8\cos(2(\theta - \theta'))]_+$, $M_{IE} = 13.3/\pi\rho_\theta$, and $M_{II} = 0$. (After Li and Dayan, 1999.)

The network oscillates in response to either constant or tuned input. Figure 7.23A shows the time average of the oscillating activities of the neurons in the network as a function of their preferred angles for noisy tuned (solid curve) and untuned (dashed curve) inputs. Neurons respond to the tuned input in a highly tuned and amplified manner. Despite the high degree of amplification, the average response of the neurons to untuned input is almost independent of θ. Figures 7.23B and 7.23C show the activities of individual neurons with $\theta = 0^\circ$ ("o") and $\theta = -37^\circ$ ("x") over time for the tuned and untuned inputs respectively. The network does not produce persistent perception, because the output to an untuned input is itself untuned. In contrast, a nonoscillatory version of this network, with $\tau_I = 0$, exhibits tuned sustained activity in response to an untuned input for recurrent weights this strong. The oscillatory network can thus operate in a regime of high selective amplification without generating spurious tuned activity.

7.6 Stochastic Networks

Up to this point, we have considered models in which the output of a cell is a deterministic function of its input. In this section, we introduce a network model called the Boltzmann machine, for which the input-output relationship is stochastic. Boltzmann machines are interesting from the perspective of learning, and also because they offer an alternative interpretation of the dynamics of network models.

Boltzmann machine

In the simplest form of Boltzmann machine, the neurons are treated as binary, so $v_a(t) = 1$ if unit a is active at time t (e.g., it fires a spike between times t and $t + \Delta t$ for some small value of Δt), and $v_a(t) = 0$ if it is inactive.

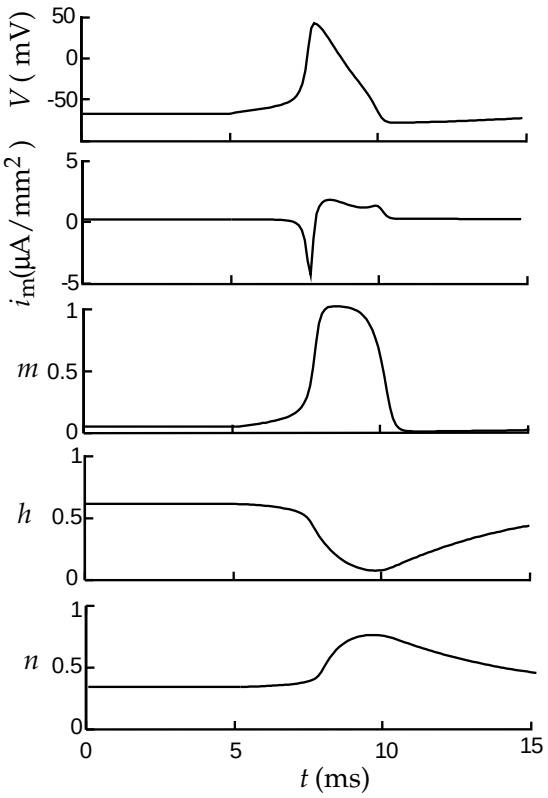

Figure 5.11 The dynamics of V, m, h, and n in the Hodgkin-Huxley model during the firing of an action potential. The upper-most trace is the membrane potential, the second trace is the membrane current produced by the sum of the Hodgkin-Huxley K^+ and Na^+ conductances, and subsequent traces show the temporal evolution of m, h, and n. Current injection was initiated at $t = 5$ ms.

membrane potential up to about -50 mV, the m variable that describes activation of the Na^+ conductance suddenly jumps from nearly 0 to a value near 1. Initially, the h variable, expressing the degree of inactivation of the Na^+ conductance, is around 0.6. Thus, for a brief period both m and h are significantly different from 0. This causes a large influx of Na^+ ions, producing the sharp downward spike of inward current shown in the second trace from the top. The inward current pulse causes the membrane potential to rise rapidly to around 50 mV (near the Na^+ equilibrium potential). The rapid increase in both V and m is due to a positive feedback effect. Depolarization of the membrane potential causes m to increase, and the resulting activation of the Na^+ conductance makes V increase. The rise in the membrane potential causes the Na^+ conductance to inactivate by driving h toward 0. This shuts off the Na^+ current. In addition, the rise in V activates the K^+ conductance by driving n toward 1. This increases the K^+ current, which drives the membrane potential back down to negative values. The final recovery involves the readjustment of m, h, and n to their initial values.

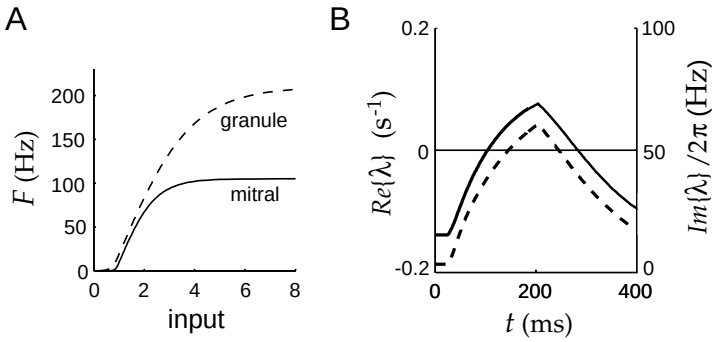

Figure 7.21 Activation functions and eigenvalues for the olfactory bulb model. (A) The activation functions F_E (solid curve) for the mitral cells, and F_I (dashed curve) for the granule cells. (B) The real (solid line, left axis) and imaginary (dashed line, right axis) parts of the eigenvalue that determines whether the network model exhibits fixed-point or oscillatory behavior. These are plotted as a function of time during a sniff cycle. When the real part of the eigenvalue becomes greater than 0, it determines the growth rate away from the fixed point, and the imaginary part divided by 2π determines the initial frequency of the resulting oscillations. (Adapted from Li, 1995.)

cells, and $\mathbf{h}_I$ is a constant representing top-down input that exists from the olfactory cortex to the granule cells.

The field potential in figure 7.20A shows oscillations during each sniff, but not between sniffs. For the model to match this pattern of activity, the input from the olfactory receptors, $\mathbf{h}_E$, must induce a transition between fixed-point and oscillatory activity. Before a sniff, the network must have a stable fixed point with low activities. As $\mathbf{h}_E$ increases during a sniff, this steady-state configuration must become unstable, leading to oscillatory activity. The analysis of the stability of the fixed point and the onset of oscillations is closely related to our previous stability analysis of the model of homogeneous populations of coupled excitatory and inhibitory neurons. It is based on properties of the eigenvalues of the linear stability matrix (see the Mathematical Appendix). In this case, the stability matrix includes contributions from the derivatives of the activation functions evaluated at the fixed point. For the fixed point to become unstable, the real part of at least one of the eigenvalues that arise in this analysis must become larger than 0. To ensure oscillations, at least one of these destabilizing eigenvalues should have a nonzero imaginary part. These requirements impose constraints on the connections between the mitral and granule cells and on the inputs.

Figure 7.21B shows the real and imaginary parts of the relevant eigenvalue, labeled λ, during one sniff cycle. About 100 ms into the cycle the real part gets bigger than 1. Reading off the imaginary part at this point, we see that this sets off roughly 40 Hz oscillations in the network. These oscillations stop about 300 ms into the sniff cycle when the real part of λ drops below 0. The input $\mathbf{h}_E$ from the receptors plays two critical roles in this process. First, it makes the real part of the eigenvalue greater than 0 by

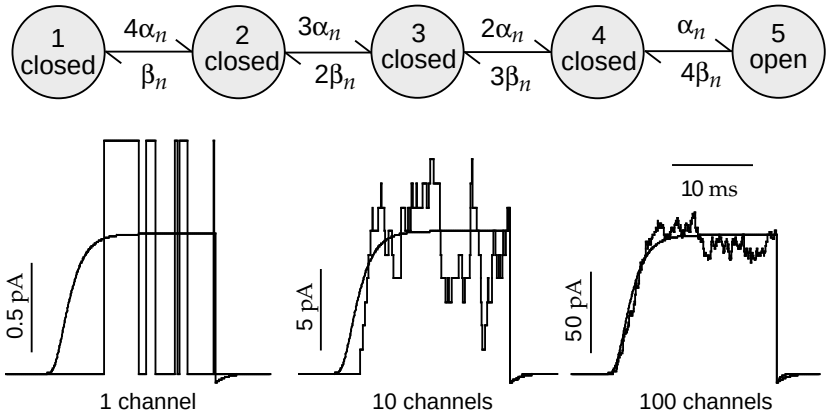

Figure 5.12 A model of the delayed-rectifier K^+ channel. The upper diagram shows the states and transition rates of the model. In the simulations shown in the lower panels, the membrane potential was initially held at -100 mV, then held at 10 mV for 20 ms, and finally returned to a holding potential of -100 mV. The smooth curves in these panels show the membrane current predicted by the Hodgkin-Huxley model in this situation. The left panel shows a simulation of a single channel that opened several times during the depolarization. The middle panel shows the total current from 10 simulated channels, and the right panel corresponds to 100 channels. As the number of channels increases, the Hodgkin-Huxley model provides a more accurate description of the current.

(scaled by the appropriate maximal conductance). For each channel, the pattern of opening and closing is random, but when enough channels are summed, the total current matches that of the Hodgkin-Huxley model quite well.

To see how the channel model in figure 5.12 reproduces the results of the Hodgkin-Huxley model when the currents from many channels are summed, we must consider a probabilistic description of the channel model. We denote the probability that a channel is in state a of figure 5.12 by p_a, with $a = 1, 2, \ldots, 5$. Dynamic equations for these probabilities are easily derived by setting the rate of change for a given p_a equal to the probability per unit time of entry into state a from other states minus the rate for leaving state a. The entry probability per unit time is the product of the appropriate transition rate times the probability that the state making the transition is occupied. The probability per unit time for leaving is p_a times the sum of all the rates for possible transitions out of the state. Following this reasoning, the equations for the state probabilities are (using the notation $\dot{p} = dp/dt$)

$$\begin{aligned}
\dot{p_1} &= \beta_n p_2 - 4\alpha_n p_1 \\
\dot{p_2} &= 4\alpha_n p_1 + 2\beta_n p_3 - (\beta_n + 3\alpha_n) p_2 \\
\dot{p_3} &= 3\alpha_n p_2 + 3\beta_n p_4 - (2\beta_n + 2\alpha_n) p_3 \\
\dot{p_4} &= 2\alpha_n p_3 + 4\beta_n p_5 - (3\beta_n + \alpha_n) p_4 \\
\dot{p_5} &= \alpha_n p_4 - 4\beta_n p_5 \,.
\end{aligned} \tag{5.26}$$

A solution for these equations can be constructed if we recall that, in the

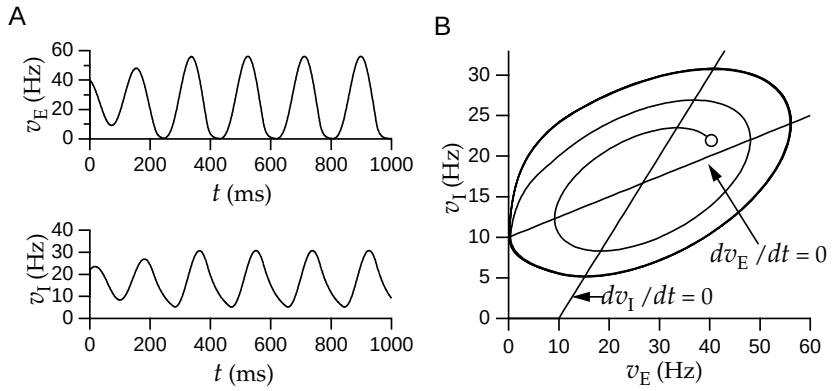

Figure 7.19 Activity of the excitatory-inhibitory firing-rate model when the fixed point is unstable. (A) The excitatory and inhibitory firing rates settle into periodic oscillations. (B) The phase-plane trajectory is a counterclockwise spiral that joins the limit cycle, which is the closed orbit. The open circle marks the initial values $v_E(0)$ and $v_I(0)$. For this example, $\tau_I = 50$ ms.

in figure 7.17B. This figure indicates that the fixed point is stable if $\tau_I < 40$ ms and unstable for larger values of τ_I.

Figures 7.18 and 7.19 show examples in which the fixed point is stable and unstable, respectively. In figure 7.18A, the oscillations in v_E and v_I are damped, and the firing rates settle down to the stable fixed point. The corresponding phase-plane trajectory is a collapsing spiral (figure 7.18B). In figure 7.19A the oscillations grow, and in figure 7.19B the trajectory is a spiral that expands outward until the system enters a limit cycle. A limit cycle is a closed orbit in the phase plane indicating periodic behavior. The fixed point is unstable in this case, but the limit cycle is stable. Without rectification, the phase-plane trajectory would spiral out from the unstable fixed point indefinitely. The rectification nonlinearity prevents the spiral trajectory from expanding past 0 and thereby stabilizes the limit cycle.

limit cycle

There are a number of ways that a nonlinear system can make a transition from a stable fixed point to a limit cycle. Such transitions are called bifurcations. The transition seen between figures 7.18 and 7.19 is a Hopf bifurcation. In this case, a fixed point becomes unstable as a parameter is changed (in this case τ_I) when the real part of a complex eigenvalue changes sign. In a Hopf bifurcation, the limit cycle emerges at a finite frequency, which is similar to the behavior of a type II neuron when it starts firing action potentials, as discussed in chapter 6. Other types of bifurcations produce type I behavior with oscillations emerging at 0 frequency (chapter 6). One example of this is a saddle-node bifurcation, which occurs when parameters are changed such that two fixed points, one stable and one unstable, meet at the same point in the phase plane.

Hopf bifurcation

saddle-node bifurcation

identical to that of the Hodgkin-Huxley model, with transition rates determined by Hodgkin-Huxley functions $\alpha_m(V)$ and $\beta_m(V)$ and appropriate combinatoric factors. State 4 is the open state. The transition to the inactivated state 5, however, is quite different from the inactivation process in the Hodgkin-Huxley model. Inactivation transitions to state 5 can occur only from states 2, 3, and 4, and the corresponding transition rates k_1, k_2, and k_3 are constants, independent of voltage. The deinactivation process occurs at the Hodgkin-Huxley rate $\alpha_h(V)$ from state 5 to state 3.

Figure 5.13 shows simulations of this Na^+ channel model. In contrast to the K^+ channel model shown in figure 5.12, this model does not reproduce exactly the results of the Hodgkin-Huxley model when large numbers of channels are summed. Nevertheless, the two models agree quite well, as seen in the lower right panel of figure 5.13. The agreement, despite the different mechanisms of inactivation, is due to the speed of the activation process for the Na^+ conductance. The inactivation rate function $\beta_h(V)$ in the Hodgkin-Huxley model has a sigmoidal form similar to the asymptotic activation function $m_\infty(V)$ (see equation 5.24). This is indicative of the actual dependence of inactivation on m and not on V. However, the activation variable m of the Hodgkin-Huxley model reaches its voltage-dependent asymptotic value $m_\infty(V)$ so rapidly that it is difficult to distinguish inactivation processes that depend on m from those that depend on V. Differences between the two models are apparent only during a submillisecond time period while the conductance is activating. Experiments that can resolve this time scale support the channel model over the original Hodgkin-Huxley description.

5.8 Synaptic Conductances

Synaptic transmission at a spike-mediated chemical synapse begins when an action potential invades the presynaptic terminal and activates voltage-dependent Ca^{2+} channels, leading to a rise in the concentration of Ca^{2+} within the terminal. This causes vesicles containing transmitter molecules to fuse with the cell membrane and release their contents into the synaptic cleft between the pre- and postsynaptic sides of the synapse. The transmitter molecules then diffuse across the cleft and bind to receptors on the postsynaptic neuron. Binding of transmitter molecules leads to the opening of ion channels that modify the conductance of the postsynaptic neuron, completing the transmission of the signal from one neuron to the other. Postsynaptic ion channels can be activated directly by binding to the transmitter, or indirectly when the transmitter binds to a distinct receptor that affects ion channels through an intracellular second-messenger signaling pathway.

As with a voltage-dependent conductance, a synaptic conductance can be written as the product of a maximal conductance and an open channel probability, $g_s = \overline{g}_s P$. The open probability for a synaptic conductance can be expressed as a product of two terms that reflect processes occurring on

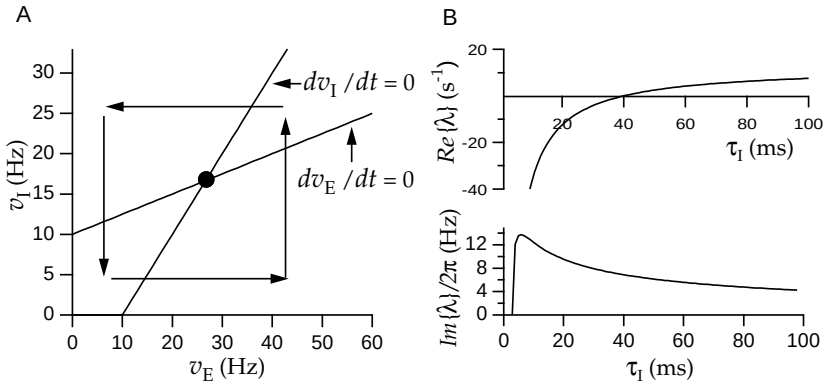

Figure 7.17 (A) Nullclines, flow directions, and fixed point for the firing-rate model of interacting excitatory and inhibitory neurons. The two straight lines are the nullclines along which $dv_E/dt = 0$ or $dv_I/dt = 0$. The filled circle is the fixed point of the model. The horizontal and vertical arrows indicate the directions that v_E (horizontal arrows) and v_I (vertical arrows) flow in different regions of the phase plane relative to the nullclines. (B) Real (upper panel) and imaginary (lower panel) parts of the eigenvalue determining the stability of the fixed point. To the left of the point where the imaginary part of the eigenvalue goes to 0, both eigenvalues are real. The imaginary part has been divided by 2π to give the frequency of oscillations near the fixed point.

dv_I/dt are positive on one side of their nullclines and negative on the other, as the reader can verify from equations 7.50 and 7.51 Above the nullcline along which $dv_E/dt = 0$, $dv_E/dt < 0$, and below it $dv_E/dt > 0$. Similarly, $dv_I/dt > 0$ to the right of the nullcline where $dv_I/dt = 0$, and $dv_I/dt < 0$ to the left of it. This determines the direction of flow in the phase plane, as denoted by the horizontal and vertical arrows in figure 7.17A. Furthermore, the rate of flow typically slows if the phase-plane trajectory approaches a nullcline.

At a fixed point of a dynamic system, the dynamic variables remain at constant values. In the model being considered, a fixed point occurs when the firing rates v_E and v_I take values that make $dv_E/dt = dv_I/dt = 0$. Because a fixed point requires both derivatives to vanish, it can occur only at an intersection of nullclines. The model we are considering has a single fixed point (at $v_E = 26.67$, $v_I = 16.67$) denoted by the filled circle in figure 7.17A. A fixed point provides a potential static configuration for the system, but it is critically important whether the fixed point is stable or unstable. If a fixed point is stable, initial values of v_E and v_I near the fixed point will be drawn toward it over time. If the fixed point is unstable, nearby configurations are pushed away from the fixed point, and the system will remain at the fixed point indefinitely only if the rates are set initially to the fixed-point values with infinite precision.

Linear stability analysis can be used to determine whether a fixed point is stable or unstable. To do this we take derivatives of the expressions for dv_E/dt and dv_I/dt obtained by dividing the right sides of equations 7.50

$GABA_A$, $GABA_B$

GABA activates two important inhibitory synaptic conductances in the brain. $GABA_A$ receptors produce a relatively fast ionotropic Cl^- conductance. $GABA_B$ receptors are metabotropic, and act to produce a slower and longer-lasting K^+ conductance.

gap junctions

In addition to chemical synapses, neurons can be coupled through electrical synapses (gap junctions) that produce a synaptic current proportional to the difference between the pre- and postsynaptic membrane potentials. Some gap junctions rectify so that positive and negative current flows are not equal for potential differences of the same magnitude.

The Postsynaptic Conductance

In a simple model of a directly activated receptor channel, the transmitter interacts with the channel through a binding reaction in which k transmitter molecules bind to a closed receptor and open it. In the reverse reaction, the transmitter molecules unbind from the receptor and it closes. These processes are analogous to the opening and closing involved in the gating of a voltage-dependent channel, and the same type of equation is used to describe how the open probability P_s changes with time,

$$\frac{dP_s}{dt} = \alpha_s(1 - P_s) - \beta_s P_s \,. \tag{5.27}$$

Here, β_s determines the closing rate of the channel and is usually assumed to be a constant. The opening rate, α_s, on the other hand, depends on the concentration of transmitter available for binding to the receptor. If the concentration of transmitter at the site of the synaptic channel is [transmitter], the probability of finding k transmitter molecules within binding range of the channel is proportional to [transmitter]^k, and α_s is some constant of proportionality times this factor.

When an action potential invades the presynaptic terminal, the transmitter concentration rises and α_s grows rapidly, causing P_s to increase. Following the release of transmitter, diffusion out of the cleft, enzyme-mediated degradation, and presynaptic uptake mechanisms can all contribute to a rapid reduction of the transmitter concentration. This sets α_s to 0, and P_s follows suit by decaying exponentially with a time constant $1/\beta_s$. Typically, the time constant for channel closing is considerably larger than the opening time.

As a simple model of transmitter release, we assume that the transmitter concentration in the synaptic cleft rises extremely rapidly after vesicle release, remains at a high value for a period of duration T, and then falls rapidly to 0. Thus, the transmitter concentration is modeled as a square pulse. While the transmitter concentration is nonzero, α_s takes a constant value much greater than β_s, otherwise $\alpha_s = 0$. Suppose vesicle release occurs at time $t = 0$ and that the synaptic channel open probability takes the value $P_s(0)$ at this time. While the transmitter concentration in the cleft

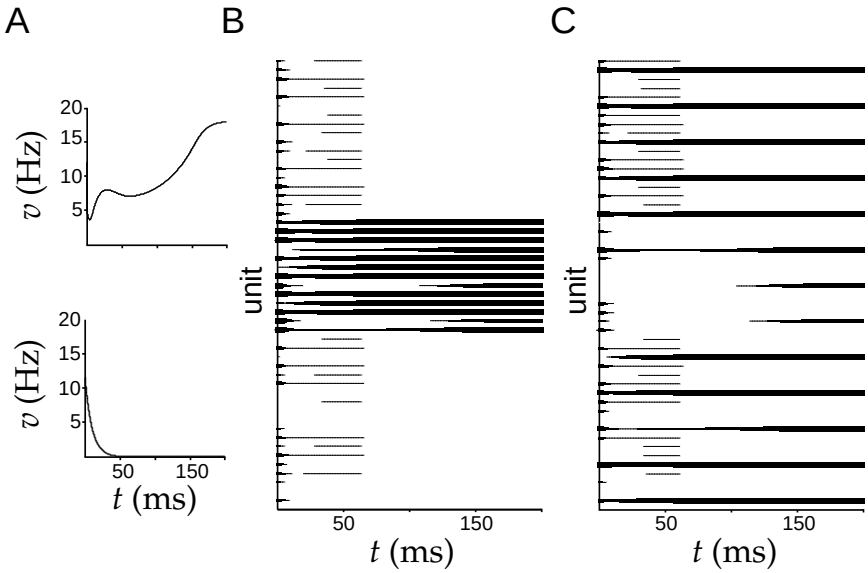

Figure 7.16 Associative recall of memory patterns in a network model. Panel A shows two representative units, and panels B and C show the firing rates of all 50 units plotted against time. The thickness of the horizontal lines in these plots is proportional to the firing rate of the corresponding neuron. (A) Firing rates of representative neurons. The upper panel shows the firing rate of one of the neurons corresponding to a nonzero component of the recalled memory pattern. The firing rate achieves a nonzero (and nonsaturated) steady-state value. The lower panel shows the firing rate of a neuron corresponding to a zero component of the recalled memory pattern. This goes to 0. (B) Recall of one of the stored memory patterns. The stored pattern had nonzero values for units 18 through 31. The initial state of the network was random, but with a bias toward this particular pattern. The final state is similar to the memory pattern. (C) Recall of another of the stored memory patterns. The stored pattern had nonzero values for every fourth unit. The initial state of the network was again random, but biased toward this pattern. The final state is similar to the memory pattern. This model uses the matrix of equation 7.48 with $\alpha = 0.25$ and $\lambda = 1.25$, and the activation function $F(I) = 150\text{ Hz}[\tanh((I + 20\text{ Hz})/(150\text{ Hz}))]_+$.

7.5 Excitatory-Inhibitory Networks

In this section, we discuss models in which excitatory and inhibitory neurons are described separately by equations 7.12 and 7.13. These models exhibit richer dynamics than the single population models with symmetric coupling matrices we have analyzed up to this point. In models with excitatory and inhibitory subpopulations, the full synaptic weight matrix is not symmetric, and network oscillations can arise. We begin by analyzing a model of homogeneous coupled excitatory and inhibitory populations. We introduce methods for determining whether this model exhibits constant or oscillatory activity. We then present two network models in which oscillations appear. The first is a model of the olfactory bulb, and the second displays selective amplification in an oscillatory mode.

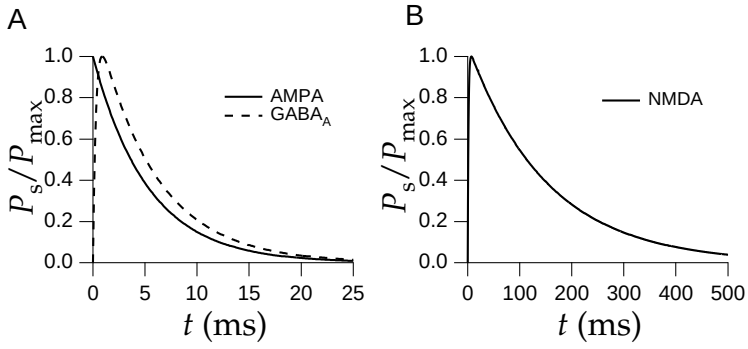

Figure 5.15 Time-dependent open probabilities fitted to match AMPA, $GABA_A$, and NMDA synaptic conductances. (A) The AMPA curve is a single exponential described by equation 5.31 with τ_s = 5.26 ms. The $GABA_A$ curve is a difference of exponentials with τ_1 = 5.6 ms and τ_{rise} = 0.3 ms. (B) The NMDA curve is the differences of two exponentials with τ_1 = 152 ms and τ_{rise} = 1.5 ms. (Parameters are from Destexhe et al., 1994.)

sequence of action potentials at arbitrary times can be modeled by allowing P_s to decay exponentially to 0 according to the equation

$$\tau_s \frac{dP_s}{dt} = -P_s\,, \tag{5.31}$$

and, on the basis of the equation 5.30, making the replacement

$$P_s \rightarrow P_s + P_{max}(1 - P_s) \tag{5.32}$$

immediately after each presynaptic action potential.

Equations 5.28 and 5.29 can also be used to model synapses with slower rise times, but other functional forms are often used. One way of describing both the rise and the fall of a synaptic conductance is to express P_s as the difference of two exponentials (see the $GABA_A$ and NMDA traces in figure 5.15). For an isolated presynaptic action potential occurring at $t = 0$, the synaptic conductance is written as

$$P_s = P_{max} B \left(\exp(-t/\tau_1) - \exp(-t/\tau_2)\right)\,, \tag{5.33}$$

where $\tau_1 > \tau_2$, and B is a normalization factor that assures that the peak value of P_s is equal to P_{max},

$$B = \left(\left(\frac{\tau_2}{\tau_1}\right)^{\tau_{rise}/\tau_1} - \left(\frac{\tau_2}{\tau_1}\right)^{\tau_{rise}/\tau_2}\right)^{-1}. \tag{5.34}$$

The rise time of the synapse is determined by $\tau_{rise} = \tau_1\tau_2/(\tau_1 - \tau_2)$, while the fall time is set by τ_1. This conductance reaches its peak value $\tau_{rise}\ln(\tau_1/\tau_2)$ after the presynaptic action potential.

Another way of describing a synaptic conductance is to use the expression

$$P_s = \frac{P_{max} t}{\tau_s}\exp(1 - t/\tau_s) \tag{5.35}$$

for any memory pattern $\mathbf{v}^m$. The second term on the right side follows from the fact that $\mathbf{n} \cdot \mathbf{v}^m = c\alpha N_v$. Treated component by component, equation 7.42 for this matrix separates into two conditions: one for the components of $\mathbf{v}^m$ that are 0 and another for the components of $\mathbf{v}^m$ equal to c,

$$F(-c) = 0 \quad \text{and} \quad c = F(c(\lambda - 1)) . \tag{7.46}$$

It is relatively easy to find conditions for which these equations have a solution. For positive c, the first condition is automatically satisfied for a rectifying activation function, $F(I) = 0$ for $I \leq 0$. For such a function satisfying $F'(I) > 0$ for all positive I, the second equation will be satisfied and equation 7.11 will have a stable fixed-point solution with $c > 0$ if, for example, $\lambda > 1$, $(\lambda - 1)F'(0) > 1$, and $F(c(\lambda - 1))$ grows more slowly than c for large c.

The existence of spurious fixed points decreases the usefulness of a network associative memory. This might seem to be a problem in the example we are discussing because the degeneracy of the eigenvalues means that any linear combination of memory patterns also satisfies equation 7.43. However, the nonlinearity in the network can prevent linear combinations of memory patterns from satisfying equation 7.42, even if they satisfy equation 7.43, thereby eliminating at least some of the spurious fixed points.

The problem of constructing an associative memory network thus reduces to finding the matrix $\mathbf{K}$ of equation 7.43, or at least constructing a matrix with similar properties. Because the choice of active units in each memory pattern is independent, the probability that a given unit is active in two different memory patterns is α^2. Thus, $\mathbf{v}^n \cdot \mathbf{v}^m \approx \alpha^2 c^2 N_v$ if $m \neq n$. Consider the dot product of one of the memory patterns, $\mathbf{v}^m$, with the vector $\mathbf{v}^n - \alpha c\mathbf{n}$, for some value of n. If $m = n$, $(\mathbf{v}^n - \alpha c\mathbf{n}) \cdot \mathbf{v}^m = c^2\alpha N_v(1 - \alpha)$, whereas if $m \neq n$, $(\mathbf{v}^n - \alpha c\mathbf{n}) \cdot \mathbf{v}^m \approx c^2 N_v(\alpha^2 - \alpha^2) = 0$. It follows from these results that the matrix

$$\mathbf{K} = \frac{\lambda}{c^2\alpha N_v(1 - \alpha)} \sum_{n=1}^{N_{\text{mem}}} \mathbf{v}^n(\mathbf{v}^n - \alpha c\mathbf{n}) \tag{7.47}$$

has properties similar to those of the matrix in equation 7.43, that is, $\mathbf{K} \cdot \mathbf{v}^m \approx \lambda \mathbf{v}^m$ for all m.

Recall that the Lyapunov function in equation 7.40 guarantees that the network has fixed points only if it is bounded from below and the matrix $\mathbf{M}$ is symmetric. Bounding of the Lyapunov function can be achieved if the activation function saturates. However, the recurrent weight matrix obtained by substituting expression 7.47 into equation 7.44 is not likely to be symmetric. A symmetric form of the recurrent weight matrix can be constructed by writing

$$\mathbf{M} = \frac{\lambda}{c^2\alpha N_v(1 - \alpha)} \sum_{n=1}^{N_{\text{mem}}} (\mathbf{v}^n - \alpha c\mathbf{n})(\mathbf{v}^n - \alpha c\mathbf{n}) - \frac{\mathbf{nn}}{\alpha N_v} . \tag{7.48}$$

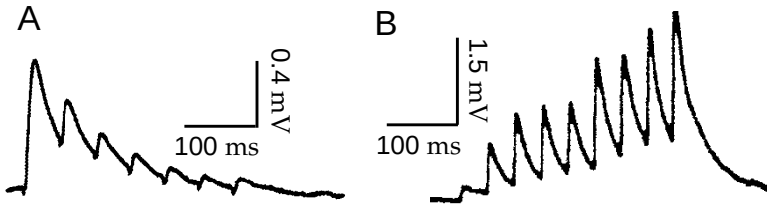

Figure 5.17 Depression and facilitation of excitatory intracortical synapses. (A) Depression of an excitatory synapse between two layer 5 pyramidal cells recorded in a slice of rat somatosensory cortex. Spikes were evoked by current injection into the presynaptic neuron, and postsynaptic potentials were recorded with a second electrode. (B) Facilitation of an excitatory synapse from a pyramidal neuron to an inhibitory interneuron in layer 2/3 of rat somatosensory cortex. (A from Markram and Tsodyks, 1996; B from Markram et al., 1998.)

Release Probability and Short-Term Plasticity

short-term plasticity

long-term plasticity

The probability of transmitter release and the magnitude of the resulting conductance change in the postsynaptic neuron can depend on the history of activity at a synapse. The effects of activity on synaptic conductances are termed short- and long-term. Short-term plasticity refers to a number of phenomena that affect the probability that a presynaptic action potential opens postsynaptic channels and that last anywhere from milliseconds to tens of seconds. The effects of long-term plasticity are extremely persistent, lasting, for example, as long as the preparation being studied can be kept alive. The modeling and implications of long-term plasticity are considered in chapter 8. Here we present a simple way of describing short-term synaptic plasticity as a modification in the release probability for synaptic transmission. Short-term modifications of synaptic transmission can involve other mechanisms than merely changes in the probability of transmission, but for simplicity we absorb all these effects into a modification of the factor P_{rel} introduced previously. Thus, P_{rel} can be interpreted more generally as a presynaptic factor affecting synaptic transmission.

depression facilitation

Figure 5.17 illustrates two principal types of short-term plasticity, depression and facilitation. Figure 5.17A shows trial-averaged postsynaptic current pulses produced in one cortical pyramidal neuron by evoking a regular series of action potentials in a second pyramidal neuron presynaptic to the first. The pulses dramatically decrease in amplitude upon repeated activation of the synaptic conductance, revealing short-term synaptic depression. Figure 5.17B shows a similar series of averaged postsynaptic current pulses recorded in a cortical inhibitory interneuron when a sequence of action potentials was evoked in a presynaptic pyramidal cell. In this case, the amplitude of the pulses increases, and thus the synapse facilitates. In general, synapses can exhibit facilitation and depression over a variety of time scales, and multiple components of short-term plasticity can be found at the same synapse. To keep the discussion simple, we consider synapses that exhibit either facilitation or depression described by a single time constant.

Associative Memory

The models of memory discussed previously in this chapter store information by means of persistent activity. This is called working or short-term memory. In biological systems, persistent activity appears to play a role in retaining information over periods of seconds to minutes. Retention of long-term memories, over periods of hours to years, is thought to involve storage by means of synaptic strengths rather than persistent activity. One general idea is that synaptic weights in a recurrently connected network are set when a memory is stored so that the network can, at a later time, internally recreate the pattern of activity that represents the stored memory. In such networks, persistent activity is used to signal memory recall and to register the identity of the retrieved item, but the synaptic weights provide the long-term storage of the possible memory patterns. The pattern of activity of the units in the network at the start of memory retrieval determines which memory is recalled through its relationship to, or association with, the pattern of activity representing that memory. Such associative networks have been used to model regions of the mammalian brain implicated in various forms of memory, including area CA3 of the hippocampus and parts of the prefrontal cortex.

In an associative (or more strictly, autoassociative) memory, a partial or approximate representation of a stored item is used to recall the full item. Unlike a standard computer memory, recall in an associative memory is based on content rather than on an address. An example would be recalling every digit of a familiar phone number, given a few of its digits as an initial clue. In a network associative memory, recurrent weights are adjusted so that the network has a set of discrete fixed points identical (or very similar) to the patterns of activity that represent the stored memories. In many cases, the dynamics of the network are governed by a Lyapunov function (equation 7.40), ensuring the existence of fixed points. Provided that not too many memories are stored, these fixed points can perfectly, or at least closely, match the memory patterns. During recall, an associative memory network performs the computational operation of pattern matching by finding the fixed-point that most closely matches the initial state of the network. Each memory pattern has a basin of attraction, defined as the set of initial activities from which the network evolves to that particular fixed point. These basins of attraction define the pattern-matching properties of the network.

Associative memory networks can be constructed from units with either continuous-valued or binary (typically on or off) activities. We consider a network of continuous-valued units described by equation 7.11 with $\mathbf{h}=\mathbf{0}$. To use this model for memory storage, we define a set of memory patterns, denoted by $\mathbf{v}^m$ with $m = 1, 2, \ldots, N_{\text{mem}}$, that we wish to store and recall. Note that $\mathbf{v}^m$ does not signify a component of a vector, but rather an entire vector identified by the superscript m. Associative recall is achieved by starting the network in an initial state that is similar to one of the memory patterns. That is, $\mathbf{v}(0) \approx \mathbf{v}^m$ for one of the m values, where "approximately

memory patterns $\mathbf{v}^m$

number of memories N_{mem}

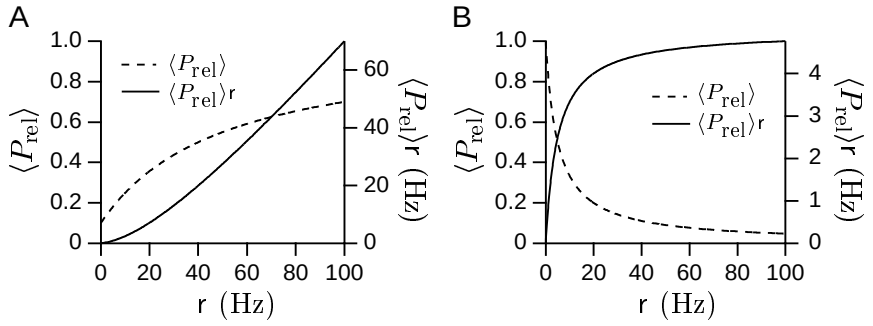

Figure 5.18 The effects of facilitation and depression on synaptic transmission. (A) Release probability and transmission rate for a facilitating synapse as a function of the firing rate of a Poisson presynaptic spike train. The dashed curve shows the rise of the average release probability as the presynaptic rate increases. The solid curve is the average rate of transmission, which is the average release probability times the presynaptic firing rate. The parameters of the model are $P_0 = 0.1$, $f_{\rm F} = 0.4$, and $\tau_{\rm P} = 50$ ms. (B) Same as A, but for the case of depression. The parameters of the model are $P_0 = 1$, $f_{\rm D} = 0.4$, and $\tau_{\rm P} = 500$ ms.

Solving for $\langle P_{\rm rel}\rangle$ gives

$$\langle P_{\rm rel}\rangle = \frac{P_0 + f_{\rm F} {\rm r} \tau_{\rm P}}{1 + {\rm r} f_{\rm F} \tau_{\rm P}} . \tag{5.40}$$

This equals P_0 at low rates and rises toward the value 1 at high rates (figure 5.18A). As a result, isolated spikes in low-frequency trains are transmitted with lower probability than spikes occurring within high-frequency bursts. The synaptic transmission rate when the presynaptic neuron is firing at rate r is the firing rate times the release probability. This is approximately P_0r for small rates and approaches r at high rates (figure 5.18A).

The value of $\langle P_{\rm rel}\rangle$ for a Poisson presynaptic spike train can also be computed in the case of depression. The only difference from the above derivation is that following a presynaptic spike, $\langle P_{\rm rel}\rangle$ is decreased to $f_{\rm D}\langle P_{\rm rel}\rangle$. Thus, the consistency condition 5.39 is replaced by

$$\langle P_{\rm rel}\rangle = P_0 + (f_{\rm D}\langle P_{\rm rel}\rangle - P_0)\frac{{\rm r}\tau_{\rm P}}{1 + {\rm r}\tau_{\rm P}} , \tag{5.41}$$

giving

$$\langle P_{\rm rel}\rangle = \frac{P_0}{1 + (1 - f_{\rm D}){\rm r}\tau_{\rm P}} . \tag{5.42}$$

This equals P_0 at low rates and decreases as $1/{\rm r}$ at high rates (figure 5.18B), which has some interesting consequences. As noted above, the average rate of successful synaptic transmissions is equal to $\langle P_{\rm rel}\rangle$ times the presynaptic rate r. Because $\langle P_{\rm rel}\rangle$ is proportional to $1/{\rm r}$ at high rates, the average transmission rate is independent of r in this range. This can be seen by the flattening of the solid curve in figure 5.18B. As a result, synapses

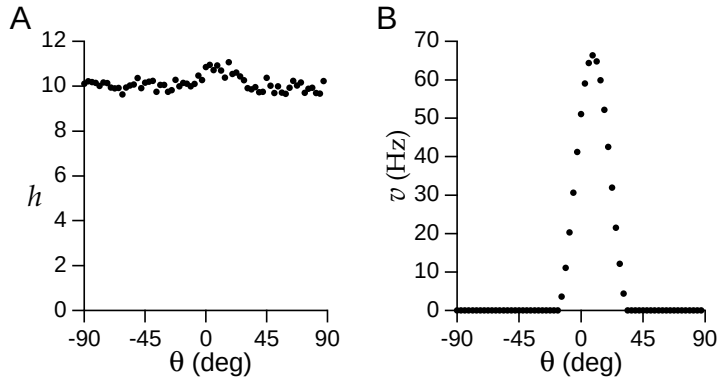

Figure 7.15 Recoding by a network model. (A) The noisy initial inputs $h(\theta)$ to 64 network neurons are shown as dots. The standard deviation of the noise is 0.25 Hz. After a short settling time, the input is set to a constant value of $h(\theta) = 10$. (B) The smooth activity profile that results from the recurrent interactions. The network model was similar to that used in figure 7.9, except that the recurrent synaptic weights were in the form of a Gabor-like function rather than a cosine, and the recurrent connections had short-range excitation and long-range inhibition. (see Pouget et al., 1998.)

timal characteristics, the network can approximate maximum likelihood decoding. Once the activity of the population of neurons has stabilized to its stereotyped shape, a simple decoding method such as vector decoding (see chapter 3) can be applied to extract the estimated value of Θ. This allows the accuracy of a vector decoding method to approach that of more complex optimal methods, because the computational work of curve fitting has been performed by the nonlinear recurrent interactions.

Figure 7.15 shows how this idea works in a network of 64 neurons receiving inputs that have Gaussian (rather than cosine) tuning curves as a function of Θ. Vector decoding applied to the reconstruction of Θ from the activity of the network or its inputs turns out to be almost unbiased. The way to judge decoding accuracy is therefore to compute the standard deviation of the decoded Θ values (chapter 3). The noisy input activity in figure 7.15A shows a slight bump around the value $\theta = 10^\circ$. Vector decoding applied to input activities with this level of noise gives a standard deviation in the decoded angle of 4.5°. Figure 7.15B shows the output of the network obtained by starting with initial activities $v(\theta) = 0$ and input $h(\theta)$ as in figure 7.15A, and then setting $h(\theta)$ to a constant (θ-independent) value to maintain sustained activity. This generates a smooth pattern of sustained population activity. Vector decoding applied to the output activities generated in this way gives a standard deviation in the decoded angle of 1.7°. This is not too far from the Cramér-Rao bound, which gives the maximum possible accuracy for any unbiased decoding scheme applied to this system (see chapter 3), which is 0.88°.

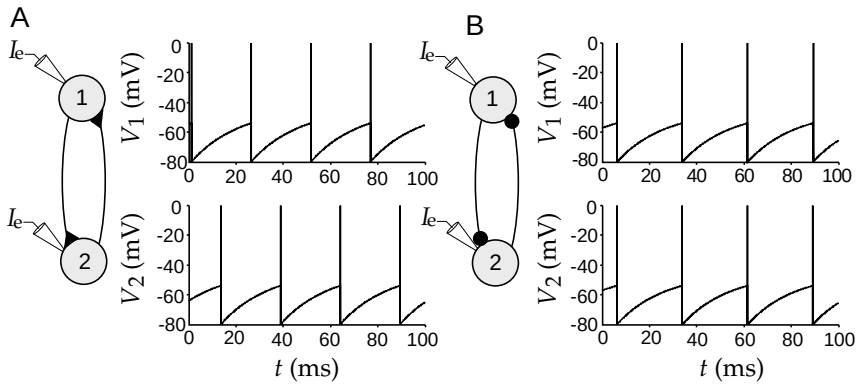

Figure 5.20 Two synaptically coupled integrate-and-fire neurons. (A) Excitatory synapses (E_s = 0 mV) produce an alternating, out-of-phase pattern of firing. (B) Inhibitory synapses (E_s = -80 mV) produce synchronous firing. Both model neurons have E_L = -70 mV, V_{th} = -54 mV, V_{reset} = -80 mV, τ_m = 20 ms, $r_m\overline{g}_s$ = 0.05, P_{max} = 1, $R_m I_e$ = 25 mV, and τ_s = 10 ms.

5.9 Synapses on Integrate-and-Fire Neurons

Synaptic inputs can be incorporated into an integrate-and-fire model by including synaptic conductances in the membrane current appearing in equation 5.8,

$$\tau_m \frac{dV}{dt} = E_L - V - r_m \overline{g}_s P_s (V - E_s) + R_m I_e \,. \tag{5.43}$$

For simplicity, we assume that $P_{rel} = 1$ in this example. The synaptic current is multiplied by r_m in equation 5.43 because equation 5.8 was multiplied by this factor. To model synaptic transmission, P_s changes whenever the presynaptic neuron fires an action potential using one of the schemes described previously.

synchronous and asynchronous firing

Figures 5.20A and 5.20B show examples of two integrate-and-fire neurons driven by electrode currents and connected by identical excitatory or inhibitory synapses. The synaptic conductances in this example are described by the α function model. This means that the synaptic conductance a time t after the occurrence of a presynaptic action potential is given by equation 5.35. The figure shows a nonintuitive effect. When the synaptic time constant is sufficiently long (τ_s = 10 ms in this example), excitatory connections produce a state in which the two neurons fire alternately, out of phase with one another, while inhibitory synapses produce synchronous firing. It is normally assumed that excitation produces synchrony. Actually, in some cases inhibitory connections can be more effective than excitatory connections at synchronizing neuronal firing.

Synapses have multiple effects on their postsynaptic targets. In equation 5.43, the term $r_m\overline{g}_s P_s E_s$ acts as a source of current to the neuron, while

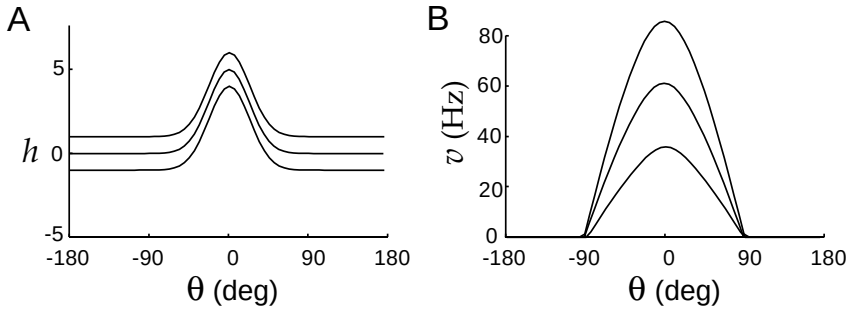

Figure 7.13 Effect of adding a constant to the input of a nonlinear recurrent network. (A) The input to the network consists of a single peak to which a constant factor has been added. (B) The gain-modulated output of the nonlinear network. The three curves correspond to the three input curves in panel A, in the same order. The model is the same as that used in figures 7.9 and 7.12.

so a constant positive input raises (and a negative input lowers) the peak of the response curve without broadening the base of the curve.

Sustained Activity

The effects illustrated in figures 7.12 and 7.13 arise because the nonlinear recurrent network has a stereotyped pattern of activity that is largely determined by interactions with other neurons in the network rather than by the feedforward input. If the recurrent connections are strong enough, the pattern of population activity, once established, can become independent of the structure of the input. For example, the recurrent network we have been studying can support a pattern of activity localized around a given preferred stimulus value, even when the input is uniform. This is seen in figure 7.14. The neurons of the network initially receive inputs that depend on their preferred angles, as seen in figure 7.14A. This produces a localized pattern of network activity (figure 7.14B). When the input is switched to the same constant value for all neurons (figure 7.14C), the network activity does not become uniform. Instead, it stays localized around the value $\theta = 0$ (figure 7.14D). This means that constant input can maintain a state that provides a memory of previous localized input activity. Networks similar to this have been proposed as models of sustained activity in the head-direction system of the rat and in prefrontal cortex during tasks involving working memory.

This memory mechanism is related to the integration seen in the linear model of eye position maintenance discussed previously. The linear network has an eigenvector $\mathbf{e}_1$ with eigenvalue $\lambda_1 = 1$. This allows $\mathbf{v} = c_1\mathbf{e}_1$ to be a static solution of the equations of the network (7.17) in the absence of input for any value of c_1. As a result, the network can preserve any initial value of c_1 as a memory. In the case of figure 7.14, the steady-state activity in the absence of tuned input is a function of $\theta - \Theta$, for any value of the angle Θ. As a result, the network can preserve any initial value of

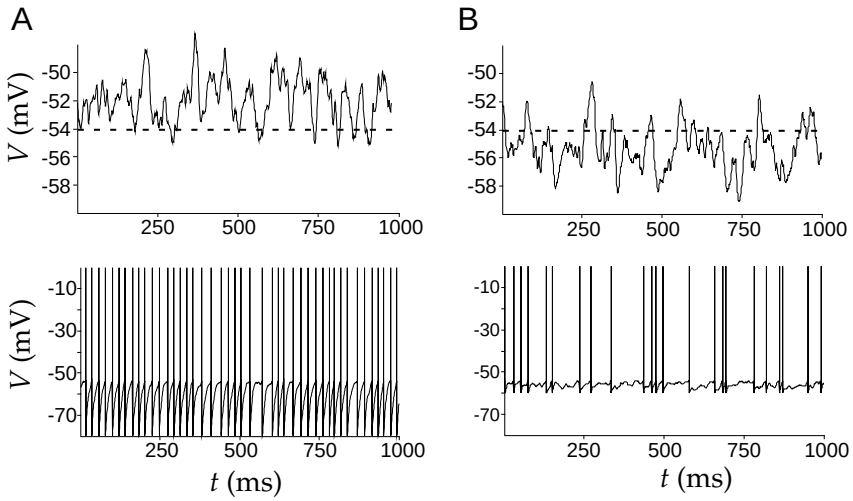

Figure 5.21 The regular and irregular firing modes of an integrate-and-fire model neuron. (A) The regular firing mode. Upper panel: The membrane potential of the model neuron when the spike generation mechanism is turned off. The average membrane potential is above the spiking threshold (dashed line). Lower panel: When the spike generation mechanism is turned on, it produces a regular spiking pattern. (B) The irregular firing mode. Upper panel: The membrane potential of the model neuron when the spike generation mechanism is turned off. The average membrane potential is below the spiking threshold (dashed line). Lower panel: When the spike generation mechanism is turned on, it produces an irregular spiking pattern. In order to keep the firing rates from differing too greatly between these two examples, the value of the reset voltage is higher in B than in A.

potential generation is blocked, more depolarized than the spiking threshold of the model (the dashed line in the figure). When the action potential mechanism is turned on (lower panel of figure 5.21A), this produces a fairly regular pattern of action potentials.

The irregularity of a spike train can be quantified using the coefficient of variation (C_V), the ratio of the standard deviation to the mean of the interspike intervals (see chapter 1). For the Poisson inputs being used in this example, $C_V = 1$, while for the spike train in the lower panel of figure 5.21A, $C_V = 0.3$. Thus, the output spike train is much more regular than the input trains. This is not surprising, because the model neuron effectively averages its many synaptic inputs. In the regular firing mode, the total synaptic input attempts to charge the neuron above the threshold, but every time the potential reaches the threshold, it gets reset and starts charging again. In this mode of operation, the timing of the action potentials is determined primarily by the charging rate of the cell, which is controlled by its membrane time constant.

Figure 5.21B shows the other mode of operation that produces an irregular firing pattern. In the irregular firing mode, the average membrane potential is more hyperpolarized than the threshold for action potential generation (upper panel of figure 5.21B). Action potentials are generated

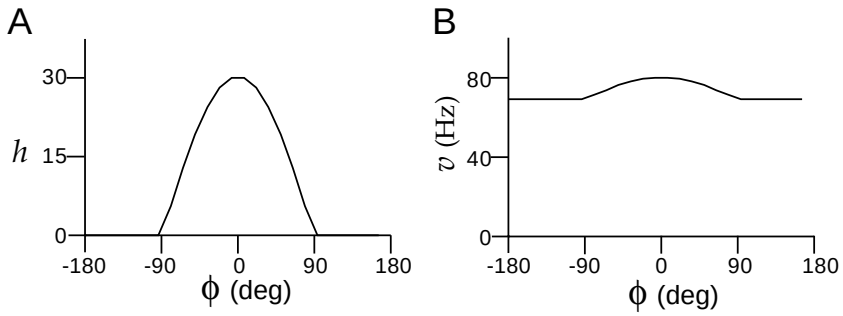

Figure 7.11 A recurrent model of complex cells. (A) The input to the network as a function of spatial phase preference. The input $h(\phi)$ is equivalent to that of a simple cell with spatial phase preference ϕ responding to a grating of 0 spatial phase. (B) Network response, which can also be interpreted as the spatial phase tuning curve of a network neuron. The network was given by equation 7.38 with $\lambda_1 = 0.95$. (Adapted from Chance et al., 1999.)

ward input. The network neurons also receive recurrent input given by the weight function $M(\phi - \phi') = \lambda_1/(2\pi\rho_\phi)$, which is the same for all connected neuron pairs. As a result, their firing rates are determined by

$$\tau_r \frac{dv(\phi)}{dt} = -v(\phi) + \left[h(\phi) + \frac{\lambda_1}{2\pi}\int_{-\pi}^{\pi} d\phi'\, v(\phi')\right]_+ . \qquad (7.38)$$

In the absence of recurrent connections ($\lambda_1 = 0$), the response of a neuron labeled by ϕ is $v(\phi) = h(\phi)$, which is equal to the response of a simple cell with preferred spatial phase ϕ. However, for λ_1 sufficiently close to 1, the recurrent model produces responses that resemble those of complex cells. Figure 7.11B shows the population response, or equivalently the single-cell response tuning curve, of the model in response to the tuned input shown in Figure 7.11A. The input, being the response of a simple cell, shows strong tuning for spatial phase. The output tuning curve, however, is almost constant as a function of spatial phase, like that of a complex cell. The spatial-phase insensitivity of the network response is due to the fact that the network amplifies the component of the input that is independent of spatial phase, because the eigenfunction of M with the largest eigenvalue is spatial-phase invariant. This changes simple cell inputs into complex cell outputs.

Winner-Takes-All Input Selection

For a linear network, the response to two superimposed inputs is simply the sum of the responses to each input separately. Figure 7.12 shows one way in which a rectifying nonlinearity modifies this superposition property. In this case, the input to the recurrent network consists of activity centered around two preferred stimulus angles, $\pm 90^\circ$. The output of the nonlinear network shown in figure 7.12B is not of this form, but instead

where t_0 is any time prior to t and $V(t_0)$ is the value of V at time t_0. Equation 5.9 is a special case of this result with $t_0 = 0$.

If I_e depends on time, the solution 5.47 is not valid. An analytic solution can still be written down in this case, but it is not particularly useful except in special cases. Over a small enough time period Δt, we can approximate $I_e(t)$ as constant and use the solution 5.47 to step from a time t to $t + \Delta t$. This requires replacing the variable t_0 in equation 5.47 with t, and t with $t + \Delta t$, so that

$$V(t + \Delta t) = V_\infty + (V(t) - V_\infty) \exp(-\Delta t/\tau_V) . \tag{5.48}$$

This equation provides an updating rule for the numerical integration of equation 5.46. Provided that Δt is sufficiently small, repeated application of the update rule 5.48 provides an accurate way of determining the membrane potential. Furthermore, this method is stable because if Δt is too large, it will only move V toward V_∞ and not, for example, make it grow without bound.

The equation for a general single-compartment conductance-based model, equation 5.6 with 5.5, can be written in the same form as equation 5.46 with

$$V_\infty = \frac{\sum_i g_i E_i + I_e/A}{\sum_i g_i} \tag{5.49}$$

and

$$\tau_V = \frac{c_m}{\sum_i g_i} . \tag{5.50}$$

Note that if c_m is in units of nF/mm^2 and the conductances are in the units μS/mm^2, τ_V comes out in ms units. Similarly, if the reversal potentials are given in units of mV, I_e is in nA, and A is in mm^2, V_∞ will be in mV units.

If we take the time interval Δt to be small enough so that the gating variables can be approximated as constant during this period, the membrane potential can again be integrated over one time step, using equation 5.48. Of course, the gating variables are not fixed, so once V has been updated by this rule, the gating variables must be updated as well.

B: Integrating the Gating Variables

All the gating variables in a conductance-based model satisfy equations of the same form,

$$\tau_z \frac{dz}{dt} = z_\infty - z , \tag{5.51}$$

where we use z to denote a generic variable. Note that this equation has the same form as equation 5.46, and it can be integrated in exactly the same way. We assume that Δt is sufficiently small so that V does not change appreciably over this time interval (and similarly [Ca^{2+}] is approximated as

The input to the model represents the orientation-tuned feedforward input arising from ON-center and OFF-center LGN cells responding to an oriented image. As a function of preferred orientation, the input for an image with orientation angle $\Theta = 0$ is

$$h(\theta) = Ac\,(1 - \epsilon + \epsilon\cos(2\theta))\ , \tag{7.37}$$

where A sets the overall amplitude and c is equal to the image contrast. The factor ϵ controls how strongly the input is modulated by the orientation angle. For $\epsilon = 0$, all neurons receive the same input, while $\epsilon = 0.5$ produces the maximum modulation consistent with a positive input. We study this model in the case when ϵ is small, which means that the input is only weakly tuned for orientation and any strong orientation selectivity must arise through recurrent interactions.

To study orientation selectivity, we want to examine the tuning curves of individual neurons in response to stimuli with different orientation angles Θ. The plots of network responses that we have been using show the firing rates $v(\theta)$ of all the neurons in the network as a function of their preferred stimulus angles θ when the input stimulus has a fixed value, typically $\Theta = 0$. As a consequence of the translation invariance of the network model, the response for other values of Θ can be obtained simply by shifting this curve so that it plots $v(\theta - \Theta)$. Furthermore, except for the asymmetric effects of noise on the input, $v(\theta - \Theta)$ is a symmetric function. These features follow from the fact that the network we are studying is invariant with respect to translations and sign changes of the angle variables that characterize the stimulus and response selectivities. An important consequence of this result is that the curve $v(\theta)$, showing the response of the entire population, can also be interpreted as the tuning curve of a single neuron. If the response of the population to a stimulus angle Θ is $v(\theta - \Theta)$, the response of a single neuron with preferred angle $\theta = 0$ is $v(-\Theta) = v(\Theta)$ from the symmetry of v. Because $v(\Theta)$ is the tuning curve of a single neuron with $\theta = 0$ to a stimulus angle Θ, the plots we show of $v(\theta)$ can be interpreted as both population responses and individual neuronal tuning curves.

Figure 7.10A shows the feedforward input to the model network for four different levels of contrast. Because the parameter ϵ was chosen to be 0.1, the modulation of the input as a function of orientation angle is small. Due to network amplification, the response of the network is much more strongly tuned to orientation (figure 7.10B). This is the result of the selective amplification of the tuned part of the input by the recurrent network. The modulation and overall height of the input curve in figure 7.10A increase linearly with contrast. The response shown in figure 7.10B, interpreted as a tuning curve, increases in amplitude for higher contrast but does not broaden. This can be seen by noting that all four curves in figure 7.10B go to 0 at the same two points. This effect, which occurs because the shape and width of the response tuning curve are determined primarily by the recurrent interactions within the network, is a feature of orientation curves of real simple cells, as seen in figure 7.10C. The width of the

based on Troyer & Miller (1997). Numerical methods for integrating the equations of neuron models are discussed in **Mascagni & Sherman (1998)**.

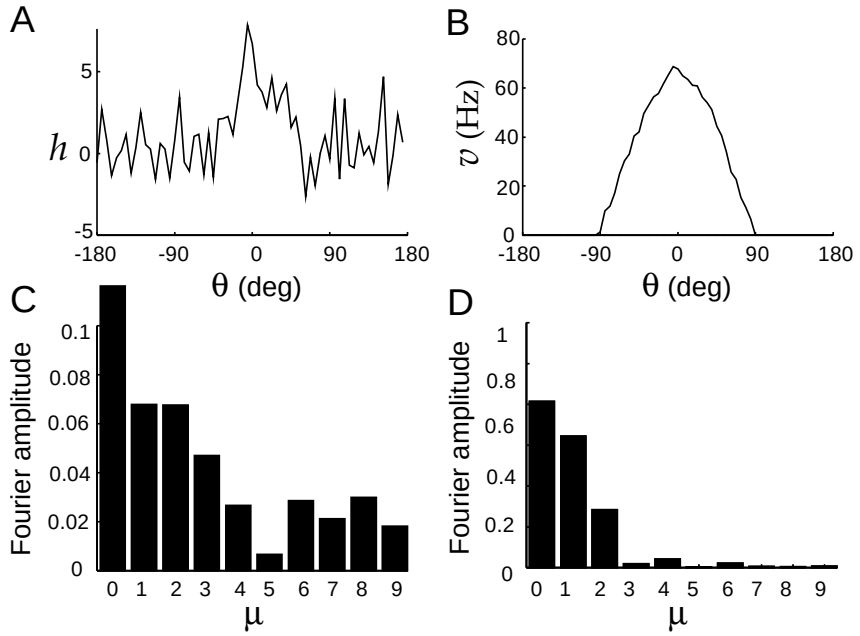

Figure 7.9 Selective amplification in a recurrent network with rectification. (A) The input $h(\theta)$ of the network plotted as a function of preferred angle. (B) The steady-state output $v(\theta)$ as a function of preferred angle. (C) Fourier transform amplitudes of the input $h(\theta)$. (D) Fourier transform amplitudes of the output $v(\theta)$. The recurrent coupling took the form of equation 7.33 with $\lambda_1 = 1.9$.

where $\boldsymbol{\gamma}$ is a vector of threshold values that we often take to be $\mathbf{0}$ (we use the notation $\mathbf{0}$ to denote a vector with all its components equal to zero). In this section, we show some examples illustrating the effect of including such a rectifying nonlinearity. Some of the features of linear recurrent networks remain when rectification is included, but several new features also appear.

vector of zeros $\mathbf{0}$

In the examples given below, we consider a continuous model, similar to that of equation 7.29, with recurrent couplings given by equation 7.33 but now including a rectification nonlinearity, so that

$$\tau_r \frac{dv(\theta)}{dt} = -v(\theta) + \left[h(\theta) + \frac{\lambda_1}{\pi} \int_{-\pi}^{\pi} d\theta' \cos(\theta - \theta') v(\theta') \right]_+ . \qquad (7.35)$$

If λ_1 is not too large, this network converges to a steady state for any constant input (we consider conditions for steady-state convergence in a later section), and therefore we often limit the discussion to the steady-state activity of the network.

Nonlinear Amplification

Figure 7.9 shows the nonlinear analog of the selective amplification shown for a linear network in figure 7.8. Once again, a noisy input (figure 7.9A)

conductance-based models, can reproduce the rich and complex dynamics of real neurons quite accurately. In this chapter, we discuss both single- and multi-compartment conductance-based models, beginning with the single-compartment case.

membrane potential equation

To review from chapter 5, the membrane potential of a single-compartment neuron model, V, is determined by integrating the equation

$$c_{\mathrm{m}}\frac{dV}{dt} = -i_{\mathrm{m}} + \frac{I_{\mathrm{e}}}{A}\,, \tag{6.1}$$

with I_{e} the electrode current, A the membrane surface area of the cell, and i_{m} the membrane current. In the following subsections, we present expressions for the membrane current in terms of the reversal potentials, maximal conductance parameters, and gating variables of the different conductances of the models being considered. The gating variables and V comprise the dynamic variables of the model. All the gating variables are determined by equations of the form

gating equations

$$\tau_z(V)\frac{dz}{dt} = z_\infty(V) - z\,, \tag{6.2}$$

where z denotes a generic gating variable. The functions $\tau_z(V)$ and $z_\infty(V)$ are determined from experimental data. For some conductances, these are written in terms of the opening and closing rates $\alpha_z(V)$ and $\beta_z(V)$ (see chapter 5), as

$$\tau_z(V) = \frac{1}{\alpha_z(V)+\beta_z(V)} \quad \text{and} \quad z_\infty(V) = \frac{\alpha_z(V)}{\alpha_z(V)+\beta_z(V)}\,. \tag{6.3}$$

We have written $\tau_z(V)$ and $z_\infty(V)$ as functions of the membrane potential, but for Ca^{2+}-dependent currents they also depend on the internal Ca^{2+} concentration. We call $\alpha_z(V)$, $\beta_z(V)$, $\tau_z(V)$, and $z_\infty(V)$ gating functions. A method for numerically integrating equations 6.1 and 6.2 is described in the appendices of chapter 5.

In the following subsections, some basic features of conductance-based models are presented in a sequence of examples of increasing complexity. We do this to illustrate the effects of various conductances and combinations of conductances on neuronal activity. Different cells (and even the same cell held at different resting potentials) can have quite different response properties due to their particular combinations of conductances. Research on conductance-based models focuses on understanding how neuronal response dynamics arises from the properties of membrane and synaptic conductances, and how the characteristics of different neurons interact when they are coupled in networks.

The Connor-Stevens Model

The Hodgkin-Huxley model of action-potential generation, discussed in chapter 5, was developed on the basis of data from the giant axon of the

where $h(\theta)$ is the feedforward input to a neuron with preferred stimulus angle θ, and we have assumed a constant density ρ_θ. Because θ is an angle, h, M, and v must all be periodic functions with period 2π. By making M a function of $\theta - \theta'$, we are imposing a symmetry with respect to translations or shifts of the angle variables. In addition, we assume that M is an even function, $M(\theta - \theta') = M(\theta' - \theta)$. This is the analog, in a continuously labeled model, of a symmetric synaptic weight matrix.

Equation 7.29 can be solved by methods similar to those used for discrete networks. We introduce eigenfunctions that satisfy

$$\rho_\theta \int_{-\pi}^{\pi} d\theta' \, M(\theta - \theta') e_\mu(\theta') = \lambda_\mu e_\mu(\theta) \, . \tag{7.30}$$

We leave it as an exercise to show that the eigenfunctions (normalized so that ρ_θ times the integral from $-\pi$ to π of their square is 1) are $1/(2\pi\rho_\theta)^{1/2}$, corresponding to $\mu = 0$, and $\cos(\mu\theta)/(\pi\rho_\theta)^{1/2}$ and $\sin(\mu\theta)/(\pi\rho_\theta)^{1/2}$ for $\mu = 1, 2, \ldots$. The eigenvalues are identical for the sine and cosine eigenfunctions and are given (including the case $\mu = 0$) by

$$\lambda_\mu = \rho_\theta \int_{-\pi}^{\pi} d\theta' \, M(\theta') \cos(\mu\theta') \, . \tag{7.31}$$

The steady-state firing rates for a constant input are given by the continuous analog of equation 7.23,

$$\begin{aligned} v_\infty(\theta) = & \frac{1}{1 - \lambda_0} \int_{-\pi}^{\pi} \frac{d\theta'}{2\pi} h(\theta') \\ & + \sum_{\mu=1}^{\infty} \frac{\cos(\mu\theta)}{1 - \lambda_\mu} \int_{-\pi}^{\pi} \frac{d\theta'}{\pi} h(\theta') \cos(\mu\theta') \\ & + \sum_{\mu=1}^{\infty} \frac{\sin(\mu\theta)}{1 - \lambda_\mu} \int_{-\pi}^{\pi} \frac{d\theta'}{\pi} h(\theta') \sin(\mu\theta') \, . \end{aligned} \tag{7.32}$$

The integrals in this expression are the coefficients in a Fourier series for the function h and are known as cosine and sine Fourier integrals (see the Mathematical Appendix). *Fourier series*

Figure 7.8 shows an example of selective amplification by a linear recurrent network. The input to the network, shown in panel A of figure 7.8, is a cosine function that peaks at 0° to which random noise has been added. Figure 7.8C shows Fourier amplitudes for this input. The Fourier amplitude is the square root of the sum of the squares of the cosine and sine Fourier integrals. No particular μ value is overwhelmingly dominant. In this and the following examples, the recurrent connections of the network are given by

$$M(\theta - \theta') = \frac{\lambda_1}{\pi\rho_\theta} \cos(\theta - \theta') \, , \tag{7.33}$$

which has all eigenvalues except λ_1 equal to 0. The network model shown in figure 7.8 has $\lambda_1 = 0.9$, so that $1/(1 - \lambda_1) = 10$. Input amplification can

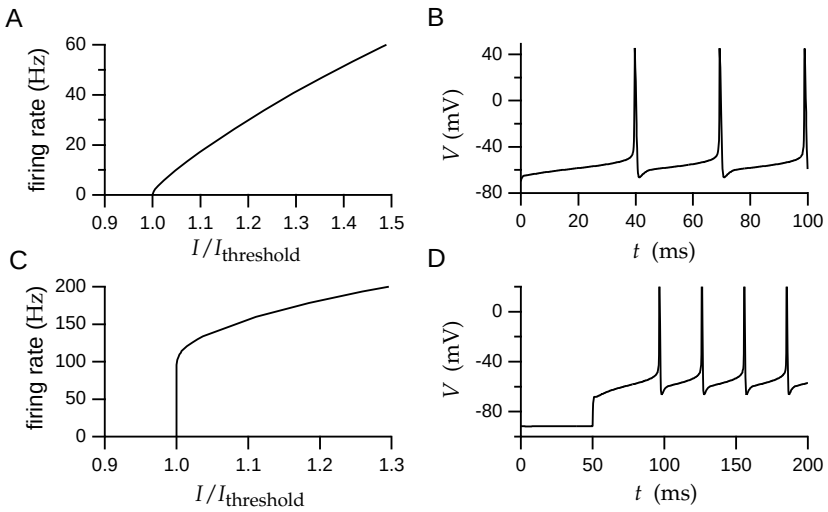

Figure 6.1 Firing of action potentials in the Connor-Stevens model. (A) Firing rate as a function of electrode current. The firing rate rises continuously from 0 as the current increases beyond the threshold value. (B) An example of action potentials generated by constant current injection. (C) Firing rate as a function of electrode current when the A-current is turned off. The firing rate now rises discontinuously from 0 as the current increases beyond the threshold value. (D) Delayed firing due to hyperpolarization. The neuron was held hyperpolarized for a prolonged period by injection of negative current. At $t = 50$ ms, the negative electrode current was switched to a positive value. The A-current delays the occurrence of the first action potential.

type I, type II

rates that rise continuously from 0 as a function of electrode current are called type I, and those with discontinuous jumps in their firing rates at threshold are called type II. An A-current is not the only mechanism that can produce a type I response but, as figures 6.1A and 6.1C show, it plays this role in the Connor-Stevens model. The Hodgkin-Huxley model produces a type II response.

Another effect of the A-current is illustrated in figure 6.1D. Here the model neuron was held hyperpolarized by negative current injection for an extended period of time, and then the current was switched to a positive value. While the neuron was hyperpolarized, the A-current deinactivated, that is, the variable b increased toward 1. When the electrode current switched sign and the neuron depolarized, the A-current first activated and then inactivated. This delayed the first spike following the change in the electrode current.

Postinhibitory Rebound and Bursting

transient Ca^{2+} *conductance*

The range of responses exhibited by the Connor-Stevens model neuron can be extended by including a transient Ca^{2+} conductance. The conductance we use was modeled by Huguenard and McCormick (1992) on the basis of

eigenvalue, $\lambda_1 = \lambda_2$, close to but less than 1. Then, equation 7.24 is replaced by

$$\mathbf{v}_\infty \approx \frac{(\mathbf{e}_1 \cdot \mathbf{h})\mathbf{e}_1 + (\mathbf{e}_2 \cdot \mathbf{h})\mathbf{e}_2}{1 - \lambda_1}\,, \tag{7.25}$$

which shows that the network now amplifies and encodes the projection of the input vector onto the plane defined by $\mathbf{e}_1$ and $\mathbf{e}_2$. In this case, the activity pattern of the network is not simply scaled when the input changes. Instead, changes in the input shift both the magnitude and the pattern of network activity. Eigenvectors that share the same eigenvalue are termed degenerate, and degeneracy is often the result of a symmetry. Degeneracy is not limited to just two eigenvectors. A recurrent network with n degenerate eigenvalues near 1 can amplify and encode a projection of the input vector from the N-dimensional space in which it is defined onto the n-dimensional subspace spanned by the degenerate eigenvectors.

Input Integration

If the recurrent weight matrix has an eigenvalue exactly equal to 1, $\lambda_1 = 1$, and all the other eigenvalues satisfy $\lambda_\nu < 1$, a linear recurrent network can act as an integrator of its input. In this case, c_1 satisfies the equation

$$\tau_r \frac{dc_1}{dt} = \mathbf{e}_1 \cdot \mathbf{h}\,, \tag{7.26}$$

obtained by setting $\lambda_1 = 1$ in equation 7.21. For arbitrary time-dependent inputs, the solution of this equation is

$$c_1(t) = c_1(0) + \frac{1}{\tau_r}\int_0^t dt'\, \mathbf{e}_1 \cdot \mathbf{h}(t')\,. \tag{7.27}$$

If $\mathbf{h}(t)$ is constant, $c_1(t)$ grows linearly with t. This explains why equation 7.24 diverges as $\lambda_1 \to 1$. Suppose, instead, that $\mathbf{h}(t)$ is nonzero for a while, and then is set to 0 for an extended period of time. When $\mathbf{h} = 0$, equation 7.22 shows that $c_\nu \to 0$ for all $\nu \neq 1$, because for these eigenvectors $\lambda_\nu < 1$. Assuming that $c_1(0) = 0$, this means that after such a period, the firing-rate vector is given, from equations 7.27 and 7.19, by

network integration

$$\mathbf{v}(t) \approx \frac{\mathbf{e}_1}{\tau_r}\int_0^t dt'\, \mathbf{e}_1 \cdot \mathbf{h}(t')\,. \tag{7.28}$$

This shows that the network activity provides a measure of the running integral of the projection of the input vector onto $\mathbf{e}_1$. One consequence of this is that the activity of the network does not cease if $\mathbf{h} = 0$, provided that the integral up to that point in time is nonzero. The network thus exhibits sustained activity in the absence of input, which provides a memory of the integral of prior input.

Networks in the brain stem of vertebrates responsible for maintaining eye position appear to act as integrators, and networks similar to the one we

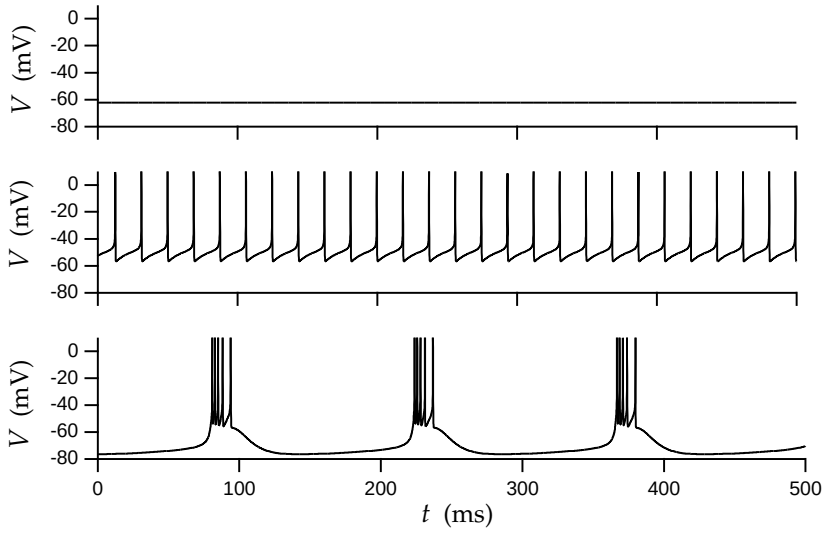

Figure 6.3 Three activity modes of a model thalamic neuron. Upper panel: with no electrode current, the model is silent. Middle panel: when a positive current is injected into the model neuron, it fires action potentials in a regular, periodic pattern. Lower panel: when negative current is injected into the model neuron, it fires action potentials in periodic bursts. (Adapted from Wang, 1994.)

action potentials. The burst in figure 6.2 is delayed due to the presence of the A-current in the Connor-Stevens model to which the Ca^{2+} conductance has been added, and it terminates when the Ca^{2+} conductance inactivates. Generation of action potentials in response to release from hyperpolarization is called postinhibitory rebound because, in a natural setting, the hyperpolarization would be caused by inhibitory synaptic input, not by current injection.

postinhibitory rebound

The transient Ca^{2+} current is an important component of models of thalamic relay neurons. These neurons exhibit different firing patterns in sleep and wakeful states. Action potentials tend to appear in bursts during sleep. Figure 6.3 shows an example of three states of activity of a model thalamic relay cell due to Wang (1994) that has, in addition to fast Na^+, delayed-rectifier K^+, and transient Ca^{2+} conductances, a hyperpolarization-activated mixed-cation conductance and a persistent Na^+ conductance. The cell is silent or fires action potentials in a regular pattern or in bursts, depending on the level of current injection. In particular, injection of small amounts of negative current leads to bursting. This occurs because the hyperpolarization due to the current injection deinactivates the transient Ca^{2+} current and activates the hyperpolarization activated current. The regular firing mode of the middle plot of figure 6.3 is believed to be relevant during wakeful states, when the thalamus is faithfully reporting input from the sensory periphery to the cortex.

thalamic relay neuron

Neurons can fire action potentials either at a steady rate or in bursts even

model is extremely useful for exploring properties of recurrent circuits, and this approach will be used both here and in the following chapters. In addition, the analysis of linear networks forms the basis for studying the stability properties of nonlinear networks. We augment the discussion of linear networks with results from simulations of nonlinear networks.

Linear Recurrent Networks

Under the linear approximation, the recurrent model of equation 7.11 takes the form

linear recurrent model

$$\tau_r \frac{d\mathbf{v}}{dt} = -\mathbf{v} + \mathbf{h} + \mathbf{M} \cdot \mathbf{v} \,. \tag{7.17}$$

Because the model is linear, we can solve analytically for the vector of output rates $\mathbf{v}$ in terms of the feedforward inputs $\mathbf{h}$ and the initial values $\mathbf{v}(0)$. The analysis is simplest when the recurrent synaptic weight matrix is symmetric, and we assume this to be the case. Equation 7.17 can be solved by expressing $\mathbf{v}$ in terms of the eigenvectors of $\mathbf{M}$. The eigenvectors $\mathbf{e}_\mu$ for $\mu = 1, 2, \ldots, N_v$ satisfy

eigenvector $\mathbf{e}$

$$\mathbf{M} \cdot \mathbf{e}_\mu = \lambda_\mu \mathbf{e}_\mu \tag{7.18}$$

for some value of the constant λ_μ, which is called the eigenvalue. For a symmetric matrix, the eigenvectors are orthogonal, and they can be normalized to unit length so that $\mathbf{e}_\mu \cdot \mathbf{e}_\nu = \delta_{\mu\nu}$. Such eigenvectors define an orthogonal coordinate system or basis that can be used to represent any N_v-dimensional vector. In particular, we can write

eigenvalue λ

eigenvector expansion

$$\mathbf{v}(t) = \sum_{\mu=1}^{N_v} c_\mu(t)\mathbf{e}_\mu \,, \tag{7.19}$$

where $c_\mu(t)$ for $\mu = 1, 2, \ldots, N_v$ are a set of time-dependent coefficients describing $\mathbf{v}(t)$.

It is easier to solve equation 7.17 for the coefficients c_μ than for $\mathbf{v}$ directly. Substituting the expansion 7.19 into equation 7.17 and using property 7.18, we find that

$$\tau_r \sum_{\mu=1}^{N_v} \frac{dc_\mu}{dt}\mathbf{e}_\mu = -\sum_{\mu=1}^{N_v}(1-\lambda_\mu)c_\mu(t)\mathbf{e}_\mu + \mathbf{h} \,. \tag{7.20}$$

The sum over μ can be eliminated by taking the dot product of each side of this equation with one of the eigenvectors, $\mathbf{e}_\nu$, and using the orthogonality property $\mathbf{e}_\mu \cdot \mathbf{e}_\nu = \delta_{\mu\nu}$ to obtain

$$\tau_r \frac{dc_\nu}{dt} = -(1-\lambda_\nu)c_\nu(t) + \mathbf{e}_\nu \cdot \mathbf{h} \,. \tag{7.21}$$

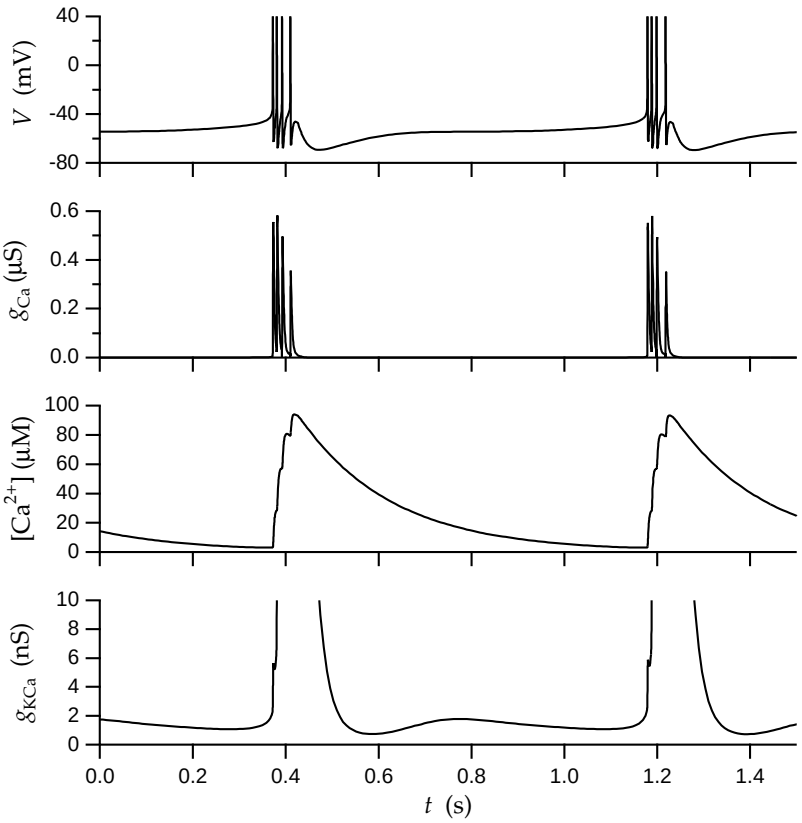

Figure 6.4 Periodic bursting in a model STG neuron. From the top, the panels show the membrane potential, the Ca^{2+} conductance, the intracellular Ca^{2+} concentration, and the Ca^{2+}-dependent K^+ conductance. The Ca^{2+}-dependent K^+ conductance is shown at an expanded scale so the reduction of the conductance due to the falling intracellular Ca^{2+} concentration during the interburst intervals can be seen. In this example, $\tau_{Ca} = 200$ ms. (Simulation by M. Goldman based on a variant of a model of Turrigiano et al., 1995, due to Z. Liu and M. Goldman.)

to the rate at which the Ca^{2+} ion concentration changes within the cell. Because the Ca^{2+} concentration is determined by dividing the number of Ca^{2+} ions in a cell by the total cellular volume and the Ca^{2+} influx is computed by multiplying i_{Ca} by the membrane surface area, γ is proportional to the surface-to-volume ratio for the cell. It also contains a factor that converts from coulombs per second of electrical current to moles per second of Ca^{2+} ions. This factor is $1/(zF)$, where z is the number of charges on the ion ($z = 2$ for Ca^{2+}) and F is the Faraday constant. If, as is normally the case, $[Ca^{2+}]$ is in moles/liter, γ should also contain a factor that converts the volume measure to liters, 10^6 mm^3/liter. Finally, γ is sometimes multiplied by an additional factor that reflects fast intracellular Ca^{2+} buffering. Most of the Ca^{2+} ions that enter a neuron are rapidly bound to intracellular buffers, so only a fraction of the Ca^{2+} current through membrane channels is actually available to change the concentration $[Ca^{2+}]$ of

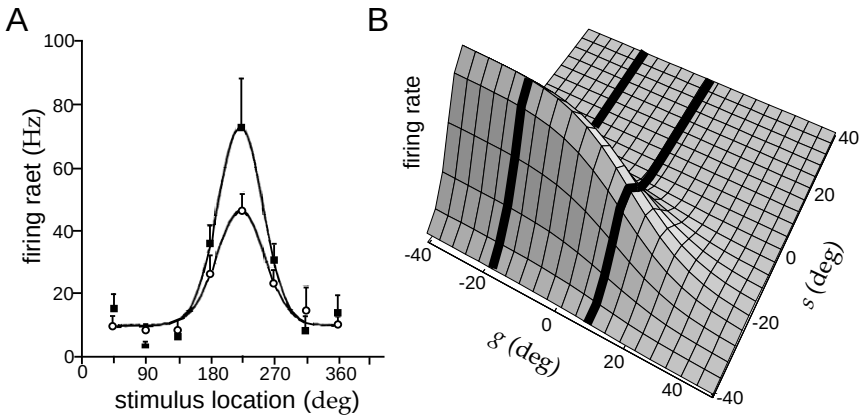

Figure 7.6 Gaze-dependent gain modulation of visual responses of neurons in posterior parietal cortex. (A) Average firing-rate tuning curves of an area 7a neuron as a function of the location of the spot of light used to evoke the response. Stimulus location is measured as an angle around a circle of possible locations on the screen and is related to, but not equal to, our stimulus variable s. The two curves correspond to the same visual images but with two different gaze directions. (B) A three-dimensional plot of the activity of a model neuron as a function of both retinal position and gaze direction. The striped bands correspond to tuning curves with different gains similar to those shown in A. (A adapted from Brotchie et al., 1995; B adapted from Pouget and Sejnowski, 1995.)

Figure 7.6B shows a mathematical description of a gain-modulated tuning curve. The response tuning curve is expressed as a product of a Gaussian function of $s-\xi$, where ξ is the preferred retinal location ($\xi = -20^\circ$ in figure 7.6B), and a sigmoidal function of $g - \gamma$, where γ is the gaze direction producing half of the maximum gain ($\gamma = 20^\circ$ in figure 7.6B). Although it does not correspond to the maximum neural response, we refer to γ as the "preferred" gaze direction.

To model a neuron with a body-centered response tuning curve, we construct a feedforward network with a single output unit representing, for example, the premotor cortex neuron shown in figure 7.5. The input layer of the network consists of a population of area 7a neurons with gain-modulated responses similar to those shown in figure 7.6B. Neurons with gains that both increase and decrease as a function of g are included in the model. The average firing rates of the input layer neurons are described by tuning curves $u = f_u(s-\xi, g-\gamma)$, with the different neurons taking different ξ and γ values.

We use continuous labeling of neurons, and replace the sum over presynaptic neurons by an integral over their ξ and γ values, inserting the appropriate density factors ρ_ξ and ρ_γ, which we assume are constant. The steady-state response of the single output neuron is determined by the continuous analog of equation 7.5. The synaptic weight from a presynaptic neuron with preferred stimulus location ξ and preferred gaze direction γ is denoted by $w(\xi, \gamma)$, so the steady-state response of the output neuron

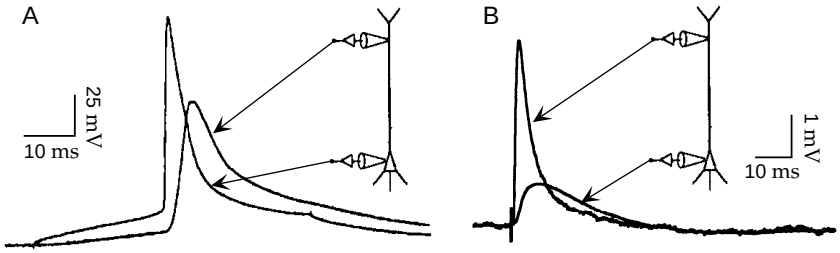

Figure 6.5 Simultaneous intracellular recordings from the soma and apical dendrite of cortical pyramidal neurons in slice preparations. (A) A pulse of current was injected into the soma of the neuron to produce the action potential seen in the somatic recording. The action potential appears delayed and with smaller amplitude in the dendritic recording. (B) A set of axon fibers was stimulated, producing an excitatory synaptic input. The excitatory postsynaptic potential (EPSP) is larger and peaks earlier in the dendrite than in the soma. Note that the scale for the potential is smaller than in A. (A adapted from Stuart and Sakmann, 1994; B adapted from Stuart and Spruston, 1998.)

across this length of cable, $\Delta V = V(x + \Delta x) - V(x)$, is then related to the amount of longitudinal current flow by Ohm's law. In chapter 5, we discussed the magnitude of this current flow, but for the present purposes, we also need to define a sign convention for its direction. We define currents flowing in the direction of increasing x as positive. By this convention, the relationship between ΔV and I_L given by Ohm's law is $\Delta V = -R_L I_L$ or $\Delta V = -r_L \Delta x I_L/(\pi a^2)$. Solving this for the longitudinal current, we find $I_L = -\pi a^2 \Delta V/(r_L \Delta x)$. It is useful to take the limit of this expression for infinitesimally short cable segments, that is, as $\Delta x \to 0$. In this limit, the ratio of ΔV to Δx becomes the derivative $\partial V/\partial x$. We use a partial derivative here because V can also depend on time. Thus, at any point along a cable of radius a and intracellular resistivity r_L, the longitudinal current flowing in the direction of increasing x is

$$I_L = -\frac{\pi a^2}{r_L}\frac{\partial V}{\partial x}. \tag{6.8}$$

The membrane potential $V(x, t)$ is determined by solving a partial differential equation, the cable equation, that describes how the currents entering, leaving, and flowing within a neuron affect the rate of change of the membrane potential. To derive the cable equation, we consider the currents within the small segment shown in figure 6.6. This segment has a radius a and a short length Δx. The rate of change of the membrane potential due to currents flowing into and out of this region is determined by its capacitance. Recall from chapter 5 that the capacitance of a membrane is determined by multiplying the specific membrane capacitance c_m by the area of the membrane. The cylinder of membrane shown in figure 6.6 has a surface area of $2\pi a \Delta x$, and hence a capacitance of $2\pi a \Delta x c_m$. The amount of current needed to change the membrane potential at a rate $\partial V/\partial t$ is thus $2\pi a \Delta x c_m \partial V/\partial t$.

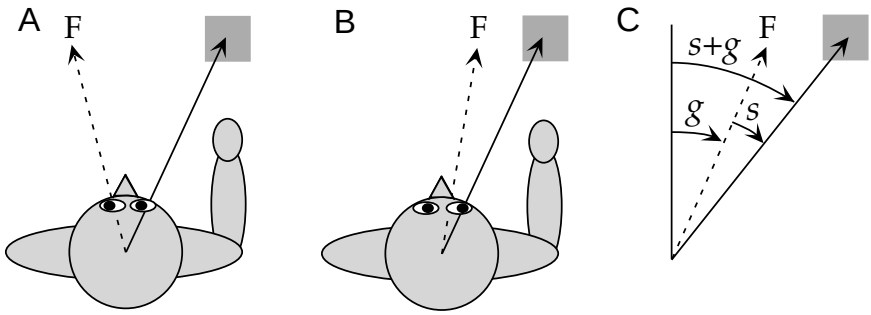

Figure 7.4 Coordinate transformations during a reaching task. (A, B) The location of the target (the gray square) relative to the body is the same in A and B, and thus the movements required to reach toward it are identical. However, the image of the object falls on different parts of the retina in A and B due to a shift in the gaze direction produced by an eye rotation that shifts the fixation point F. (C) The angles used in the analysis: s is the angle describing the location of the stimulus (the target) in retinal coordinates, that is, relative to a line directed to the fixation point; g is the gaze angle, indicating the direction of gaze relative to an axis straight out from the body. The direction of the target relative to the body is $s + g$.

7.3 Feedforward Networks

Substantial computations can be performed by feedforward networks in the absence of recurrent connections. Much of the work done on feedforward networks centers on plasticity and learning, as discussed in the following chapters. Here, we present an example of the computational power of feedforward circuits, the calculation of the coordinate transformations needed in visually guided reaching tasks.

Neural Coordinate Transformations

Reaching for a viewed object requires a number of coordinate transformations that turn information about where the image of the object falls on the retina into movement commands in shoulder-, arm-, or hand-based coordinates. To perform a transformation from retinal to body-based coordinates, information about the retinal location of an image and about the direction of gaze relative to the body must be combined. Figure 7.4A and B illustrate, in a one-dimensional example, how a rotation of the eyes affects the relationship between gaze direction, retinal location, and location relative to the body. Figure 7.4C introduces the notation we use. The angle g describes the orientation of a line extending from the head to the point of visual fixation. The visual stimulus in retinal coordinates is given by the angle s between this line and a line extending out to the target. The angle describing the reach direction, the direction to the target relative to the body, is the sum $s + g$.

Visual neurons have receptive fields fixed to specific locations on the retina. Neurons in motor areas can display visually evoked responses that

The arrow refers to the limit $\Delta x \to 0$, which we now take. We can move r_L outside the derivative in this equation under the assumption that it is not a function of position. However, the factor of a^2 must remain inside the derivative unless it is independent of x. Substituting the result 6.10 into 6.9, we obtain the cable equation,

cable equation

$$c_m \frac{\partial V}{\partial t} = \frac{1}{2ar_L} \frac{\partial}{\partial x}\left(a^2 \frac{\partial V}{\partial x}\right) - i_m + i_e \,. \tag{6.11}$$

boundary conditions for the cable equation

To determine the membrane potential, equation (6.11) must be augmented by appropriate boundary conditions. The boundary conditions specify what happens to the membrane potential when the neuronal cable branches or terminates. The point at which a cable branches, or equivalently where multiple cable segments join, is called a node. At such a branching node, the potential must be continuous, that is, the functions $V(x,t)$ defined along each of the segments must yield the same result when evaluated at the x value corresponding to the node. In addition, charge must be conserved, which means that the sum of the longitudinal currents entering (or leaving) a node along all of its branches must be 0. According to equation 6.8, the longitudinal current entering a node is proportional to the square of the cable radius times the derivative of the potential evaluated at that point, $a^2 \partial V/\partial x$. The sum of the longitudinal currents entering the node, computed by evaluating these derivatives along each cable segment at the point where they meet at the node, must be 0.

Several different boundary conditions can be imposed at the end of a terminating cable segment. One simple condition is that no current flows out of the end of the cable. By equation 6.8, this means that the spatial derivative of the potential must vanish at a termination point.

Due to the complexities of neuronal membrane currents and morphologies, the cable equation is most often solved numerically, using multicompartmental techniques described later in this chapter. However, it is useful to study analytic solutions of the cable equation in simple cases to get a feel for how different morphological features, such as long dendritic cables, branching nodes, changes in cable radii, and cable ends, affect the membrane potential.

Linear Cable Theory

Before we can solve the cable equation by any method, the membrane current i_m must be specified. We discussed models of various ion channel contributions to the membrane current in chapter 5 and earlier in this chapter. These models typically produce nonlinear expressions that are too complex to allow analytic solution of the cable equation. The analytic solutions we discuss use two rather drastic approximations: synaptic currents are ignored, and the membrane current is written as a linear function of the

It is often convenient to define the total feedforward input to each neuron in the network of figure 7.3B as $\mathbf{h} = \mathbf{W} \cdot \mathbf{u}$. Then, the output rates are determined by the equation

$$\tau_r \frac{d\mathbf{v}}{dt} = -\mathbf{v} + \mathbf{F}(\mathbf{h} + \mathbf{M} \cdot \mathbf{v}) \,. \tag{7.11}$$

Neurons are typically classified as either excitatory or inhibitory, meaning that they have either excitatory or inhibitory effects on all of their postsynaptic targets. This property is formalized in Dale's law, which states that a neuron cannot excite some of its postsynaptic targets and inhibit others. In terms of the elements of $\mathbf{M}$, this means that for each presynaptic neuron a', $M_{aa'}$ must have the same sign for all postsynaptic neurons a. To impose this restriction, it is convenient to describe excitatory and inhibitory neurons separately. The firing-rate vectors $\mathbf{v}_\mathrm{E}$ and $\mathbf{v}_\mathrm{I}$ for the excitatory and inhibitory neurons are then described by a coupled set of equations identical in form to equation 7.11,

Dale's law

excitatory-inhibitory network

$$\tau_\mathrm{E} \frac{d\mathbf{v}_\mathrm{E}}{dt} = -\mathbf{v}_\mathrm{E} + \mathbf{F}_\mathrm{E}\left(\mathbf{h}_\mathrm{E} + \mathbf{M}_\mathrm{EE} \cdot \mathbf{v}_\mathrm{E} + \mathbf{M}_\mathrm{EI} \cdot \mathbf{v}_\mathrm{I}\right) \tag{7.12}$$

and

$$\tau_\mathrm{I} \frac{d\mathbf{v}_\mathrm{I}}{dt} = -\mathbf{v}_\mathrm{I} + \mathbf{F}_\mathrm{I}\left(\mathbf{h}_\mathrm{I} + \mathbf{M}_\mathrm{IE} \cdot \mathbf{v}_\mathrm{E} + \mathbf{M}_\mathrm{II} \cdot \mathbf{v}_\mathrm{I}\right) \,. \tag{7.13}$$

There are now four synaptic weight matrices describing the four possible types of neuronal interactions. The elements of $\mathbf{M}_\mathrm{EE}$ and $\mathbf{M}_\mathrm{IE}$ are greater than or equal to 0, and those of $\mathbf{M}_\mathrm{EI}$ and $\mathbf{M}_\mathrm{II}$ are less than or equal to 0. These equations allow the excitatory and inhibitory neurons to have different time constants, activation functions, and feedforward inputs.

In this chapter, we consider several recurrent network models described by equation 7.11 with a symmetric weight matrix, $M_{aa'} = M_{a'a}$ for all a and a'. Requiring $\mathbf{M}$ to be symmetric simplifies the mathematical analysis, but it violates Dale's law. Suppose, for example, that neuron a, which is excitatory, and neuron a', which is inhibitory, are mutually connected. Then, $M_{aa'}$ should be negative and $M_{a'a}$ positive, so they cannot be equal. Equation 7.11 with symmetric $\mathbf{M}$ can be interpreted as a special case of equations 7.12 and 7.13 in which the inhibitory dynamics are instantaneous ($\tau_\mathrm{I} \to 0$) and the inhibitory rates are given by $\mathbf{v}_\mathrm{I} = \mathbf{M}_\mathrm{IE}\mathbf{v}_\mathrm{E}$. This produces an effective recurrent weight matrix $\mathbf{M} = \mathbf{M}_\mathrm{EE} + \mathbf{M}_\mathrm{EI} \cdot \mathbf{M}_\mathrm{IE}$, which can be made symmetric by the appropriate choice of the dimension and form of the matrices $\mathbf{M}_\mathrm{EI}$ and $\mathbf{M}_\mathrm{IE}$. The dynamic behavior of equation 7.11 is restricted by requiring the matrix $\mathbf{M}$ to be symmetric. For example symmetric coupling typically does not allow for network oscillations. In the latter part of this chapter, we consider the richer dynamics of models described by equations 7.12 and 7.13.

symmetric coupling

Equation 6.16 is a linear equation for v similar to the diffusion equation, and it can be solved by standard methods of mathematical analysis. The constants τ_m and λ set the scale for temporal and spatial variations in the membrane potential. For example, the membrane potential requires a time of order τ_m to settle down after a transient, and deviations in the membrane potential due to localized electrode currents decay back to 0 over a length of order λ.

The membrane potential is affected both by the form of the cable equation and by the boundary conditions imposed at branching nodes and terminations. To isolate these two effects, we consider two idealized cases: an infinite cable that does not branch or terminate, and a single branching node that joins three semi-infinite cables. Of course, real neuronal cables are not infinitely long, but the solutions we find are applicable for long cables far from their ends. We determine the potential for both of these morphologies when current is injected at a single point. Because the equation we are studying is linear, the membrane potential for any other spatial distribution of electrode current can be determined by summing solutions corresponding to current injection at different points. The use of point injection to build more general solutions is a standard method of linear analysis. In this context, the solution for a point source of current injection is called a Green's function.

Green's function

An Infinite Cable

In general, solutions to the linear cable equation are functions of both position and time. However, if the current being injected is held constant, the membrane potential settles to a steady-state solution that is independent of time. Solving for this time-independent solution is easier than solving the full time-dependent equation, because the cable equation reduces to an ordinary differential equation in the static case,

$$\lambda^2 \frac{d^2v}{dx^2} = v - r_m i_e \,. \tag{6.17}$$

For the localized current injection we wish to study, i_e is 0 everywhere except within a small region of size Δx around the injection site, which we take to be $x = 0$. Eventually we will let $\Delta x \to 0$. Away from the injection site, the linear cable equation is $\lambda^2 d^2v/dx^2 = v$, which has the general solution $v(x) = B_1 \exp(-x/\lambda) + B_2 \exp(x/\lambda)$ with as yet undetermined coefficients B_1 and B_2. These constant coefficients are determined by imposing boundary conditions appropriate to the particular morphology being considered. For an infinite cable, on physical grounds we simply require that the solution does not grow without bound when $x \to \pm\infty$. This means that we must choose the solution with $B_1 = 0$ for the region $x < 0$ and the solution with $B_2 = 0$ for $x > 0$. Because the solution must be continuous at $x = 0$, we must require $B_1 = B_2 = B$, and these two solutions can be combined into a single expression, $v(x) = B \exp(-|x|/\lambda)$. The remaining task

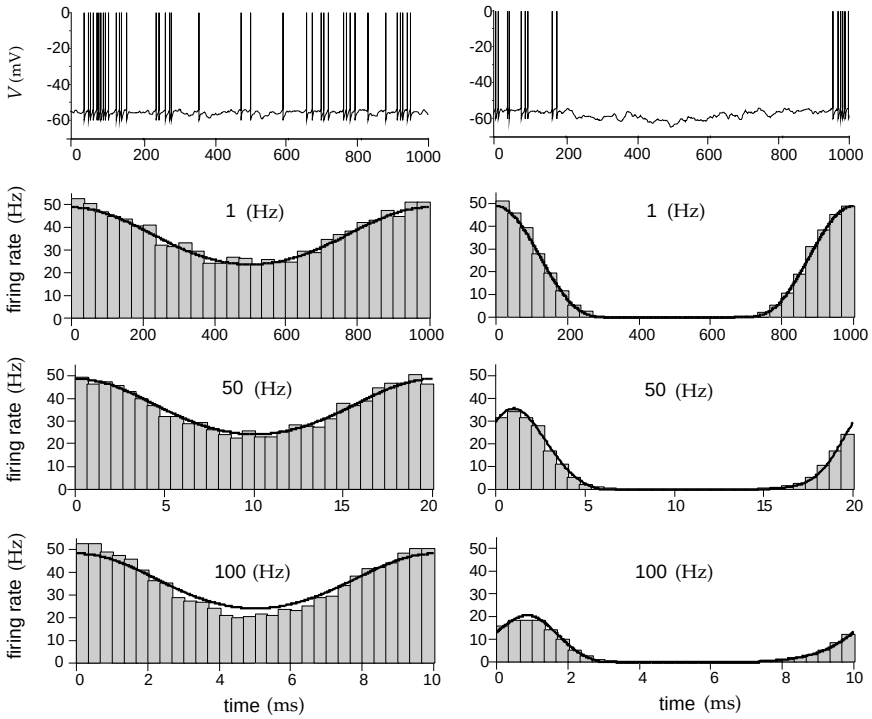

Figure 7.2 Firing rate of an integrate-and-fire neuron receiving balanced excitatory and inhibitory synaptic input and an injected current consisting of a constant and a sinusoidally varying term. For the left panels, the constant component of the injected current was adjusted so the firing never stopped during the oscillation of the varying part of the injected current. For the right panel, the constant component was lowered so the firing stopped during part of the cycle. The upper panels show two representative voltage traces of the model cell. The histograms beneath these traces were obtained by binning spikes generated over multiple cycles. They show the firing rate as a function of the time during each cycle of the injected current oscillations. The different rows correspond to 1, 50, and 100 Hz oscillation frequencies for the injected current. The solid curves show the fit of a firing-rate model that involves both instantaneous and low-pass filtered effects of the injected current. For the left panel, this reduces to the simple prediction $v = F(I(t))$. (Adapted from Chance, 2000.)

of the oscillation cycle. In this case, the firing is delayed and attenuated at high frequencies, as would be predicted by equation 7.7. In this case, the membrane potential stays below threshold for long enough periods of time that its dynamics become relevant for the firing of the neuron.

The essential message from figure 7.2 is that neither equation 7.6 nor equation 7.8 provides a completely accurate prediction of the dynamics of the firing rate at all frequencies and for all levels of injected current. A more complex model can be constructed that accurately describes the firing rate over the entire range of input current amplitudes and frequencies. The solid curves in figure 7.2 were generated by a model that expresses the firing rate as a function of both I from equation 7.6 and v from equation 7.8. In other words, it is a combination of the two models discussed in the pre-

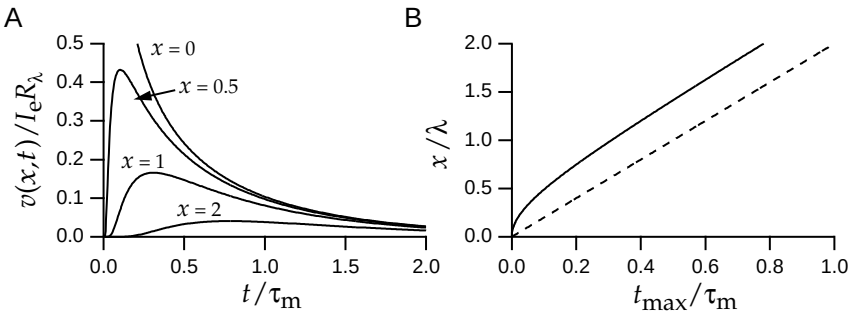

Figure 6.8 Time dependence of the potential on an infinite cable in response to a pulse of current injected at the point $x = 0$ at time $t = 0$. (A) The potential is always largest at the site of current injection. At any fixed point, it reaches its maximum value as a function of time later for measurement sites located farther away from the current source. (B) Movement of the temporal maximum of the potential. The solid line shows the relationship between the measurement location x and the time t_{max} when the potential reaches its maximum value at that location. The dashed line corresponds to a constant velocity $2\lambda/\tau_m$.

infinite cable has an input resistance of $R_\lambda/2$. Each direction of the cable acts like a resistance of R_λ, and these two act in parallel to produce a total resistance half as big. Note that each semi-infinite cable extending from the point $x = 0$ has a resistance equal to a finite cable of length λ.

We now consider the membrane potential produced by an instantaneous pulse of current injected at the point $x = 0$ at the time $t = 0$. Specifically, we consider $i_e = I_e \tau_m \delta(x)\delta(t)/2\pi a$, which means that the current pulse delivers a total charge of $I_e \tau_m$. We do not derive the solution for this case (see Tuckwell, 1988, for example), but simply state the answer,

$$v(x,t) = \frac{I_e R_\lambda}{\sqrt{4\pi t/\tau_m}} \exp\left(-\frac{\tau_m x^2}{4\lambda^2 t}\right) \exp\left(-\frac{t}{\tau_m}\right). \qquad (6.19)$$

In this case, the spatial dependence of the potential is determined by a Gaussian, rather than an exponential function. The Gaussian is always centered around the injection site, so the potential is always largest at $x = 0$. The width of the Gaussian curve around $x = 0$ is proportional to $\lambda\sqrt{t/\tau_m}$. As expected, λ sets the scale for this spatial variation, but the width also grows as the square root of the time measured in units of τ_m. The factor $(4\pi t/\tau_m)^{-1/2}$ in equation 6.19 preserves the total area under this Gaussian curve, but the additional exponential factor $\exp(-t/\tau_m)$ reduces the integrated amplitude over time. As a result, the spatial dependence of the membrane potential is described by a spreading Gaussian function with an integral that decays exponentially (figure 6.7B).

Figure 6.8 shows the solution of equation 6.19 plotted at various fixed positions as a function of time. Figure 6.8A shows that the membrane potential measured farther from the injection site reaches its maximum value at later times. It is important to keep in mind that the membrane potential spreads out from the region $x = 0$; it does not propagate like a wave. Nevertheless,

For time-independent inputs, the relation $v = F(I_s)$ is all we need to know to complete the firing-rate model. The total steady-state synaptic current predicted by equation 7.4 for time-independent $\mathbf{u}$ is $I_s = \mathbf{w} \cdot \mathbf{u}$. This generates a steady-state output firing rate $v = v_\infty$ given by

$$v_\infty = F(\mathbf{w} \cdot \mathbf{u}) . \tag{7.5}$$

The steady-state firing rate tells us how a neuron responds to constant current, but not to a current that changes with time. To model time-dependent inputs, we need to know the firing rate in response to a time-dependent synaptic current $I_s(t)$. The simplest assumption is that this is still given by the activation function, so $v = F(I_s(t))$ even when the total synaptic current varies with time. This leads to a firing-rate model in which the dynamics arises exclusively from equation 7.4,

firing-rate model with current dynamics

$$\tau_s \frac{dI_s}{dt} = -I_s + \mathbf{w} \cdot \mathbf{u} \quad \text{with} \quad v = F(I_s) . \tag{7.6}$$

An alternative formulation of a firing-rate model can be constructed by assuming that the firing rate does not follow changes in the total synaptic current instantaneously, as was assumed for the model of equation 7.6. Action potentials are generated by the synaptic current through its effect on the membrane potential of the neuron. Due to the membrane capacitance and resistance, the membrane potential is, roughly speaking, a low-pass filtered version of I_s (see the Mathematical Appendix). For this reason, the time-dependent firing rate is often modeled as a low-pass filtered version of the steady-state firing rate,

$$\tau_r \frac{dv}{dt} = -v + F(I_s(t)) . \tag{7.7}$$

The constant τ_r in this equation determines how rapidly the firing rate approaches its steady-state value for constant I_s, and how closely v can follow rapid fluctuations for a time-dependent $I_s(t)$. Equivalently, it measures the time scale over which v averages $F(I_s(t))$. The low-pass filtering effect of equation 7.7 is described in the Mathematical Appendix in the context of electrical circuit theory. The argument we have used to motivate equation 7.7 would suggest that τ_r should be approximately equal to the membrane time constant of the neuron. However, this argument really applies to the membrane potential, not the firing rate, and the dynamics of the two are not the same. Some network models use a value of τ_r that is considerably less than the membrane time constant. We re-examine this issue in the following section.

The second model that we have described involves the pair of equations 7.4 and 7.7. If one of these equations relaxes to its equilibrium point much more rapidly than the other, the pair can be reduced to a single equation. We discuss cases in which this occurs in the following section. For example, if $\tau_r \ll \tau_s$, we can make the approximation that equation 7.7 rapidly sets $v = F(I_s(t))$, and then the second model reduces to the first

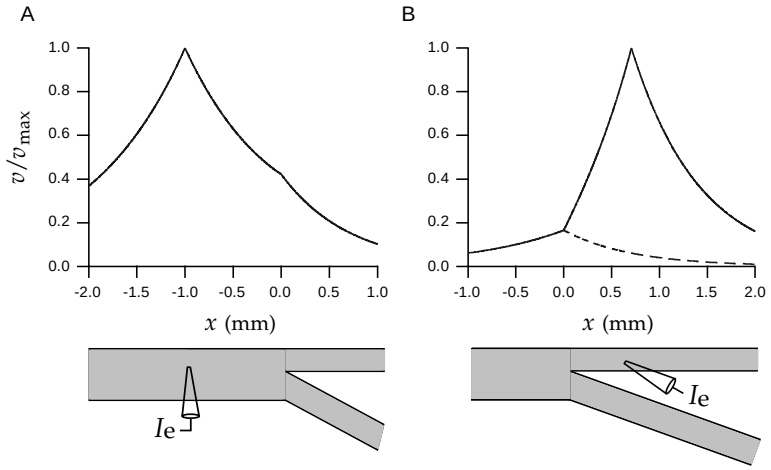

Figure 6.9 The potentials along the three branches of an isolated junction for a current injection site one electrotonic length constant away from the junction. The potential v is plotted relative to v_{max}, which is v at the site of the electrode. The thick branch has a radius of 2 μ and an electrotonic length constant $\lambda = 1$ mm, and the two thin branches have radii of 1 μ and $\lambda = 2^{-1/2}$ mm. (A) Current injection along the thick branch. The potentials along both of the thin branches, shown by the solid curve over the range $x > 0$, are identical. The solid curve over the range $x < 0$ shows the potential on the thick branch where current is being injected. (B) Current injection along one of the thin branches. The dashed line shows the potential along the thin branch where current injection does not occur. The solid line shows the potential along the thick branch for $x < 0$ and along the thin branch receiving the injected current for $x > 0$.

electrotonic length constant away from the junction. The two daughter branches have little effect on the falloff of the potential away from the electrode site in figure 6.9A. This is because the thin branches do not represent a large current sink. The thick branch has a bigger effect on the attenuation of the potential along the thin branch receiving the electrode current in figure 6.9B. This can be seen as an asymmetry in the falloff of the potential on either side of the electrode. Loading by the thick cable segment contributes to a quite severe attenuation between the two thin branches in figure 6.9B. Comparison of figures 6.9A and B reveals a general feature of static attenuation in a passive cable: attenuation near the soma due to potentials arising in the periphery is typically greater than attenuation in the periphery due to potentials arising near the soma.

The Rall Model

The infinite and semi-infinite cables we have considered are clearly mathematical idealizations. We now turn to a model neuron introduced by Rall (1959, 1977) that, though still highly simplified, captures some of the important elements that affect the responses of real neurons. Most neurons receive their synaptic inputs over complex dendritic trees. The integrated

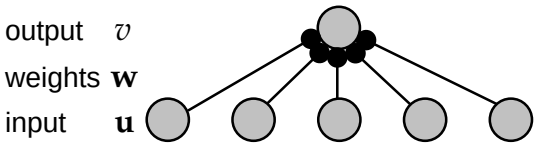

Figure 7.1 Feedforward inputs to a single neuron. Input rates $\mathbf{u}$ drive a neuron at an output rate v through synaptic weights given by the vector $\mathbf{w}$.

ering how it depends on presynaptic spikes. If an action potential arrives at input b at time 0, we write the synaptic current generated in the soma of the postsynaptic neuron at time t as $w_b K_s(t)$, where w_b is the synaptic weight and $K_s(t)$ is called the synaptic kernel. Collectively, the synaptic weights are represented by a synaptic weight vector $\mathbf{w}$, which has N_u components w_b. The amplitude and sign of the synaptic current generated by input b are determined by w_b. For excitatory synapses, $w_b > 0$, and for inhibitory synapses, $w_b < 0$. In this formulation of the effect of presynaptic spikes, the probability of transmitter release from a presynaptic terminal is absorbed into the synaptic weight factor w_b, and we do not include short-term plasticity in the model (although this can be done by making w_b a dynamic variable).

synaptic weights $\mathbf{w}$

The synaptic kernel, $K_s(t) \geq 0$, describes the time course of the synaptic current in response to a presynaptic spike arriving at time $t = 0$. This time course depends on the dynamics of the synaptic conductance activated by the presynaptic spike, and also on both the passive and the active properties of the dendritic cables that carry the synaptic current to the soma. For example, long passive cables broaden the synaptic kernel and slow its rise from 0. Cable calculations or multi-compartment simulations, such as those discussed in chapter 6, can be used to compute $K_s(t)$ for a specific dendritic structure. To avoid ambiguity, we normalize $K_s(t)$ by requiring its integral over all positive times to be 1. At this point, for simplicity, we use the same function $K_s(t)$ to describe all synapses.

synaptic kernel $K_s(t)$

Assuming that the effects of the spikes at a single synapse sum linearly, the total synaptic current at time t arising from a sequence of presynaptic spikes occurring at input b at times t_i is given by

$$w_b \sum_{t_i < t} K_s(t - t_i) = w_b \int_{-\infty}^{t} d\tau \, K_s(t - \tau) \rho_b(\tau) \, . \tag{7.1}$$

In the second expression, we have used the neural response function, $\rho_b(\tau) = \sum_i \delta(\tau - t_i)$, to describe the sequence of spikes fired by presynaptic neuron b. The equality follows from integrating over the sum of δ functions in the definition of $\rho_b(\tau)$. If there is no nonlinear interaction between different synaptic currents, the total synaptic current coming from all presynaptic inputs is obtained simply by summing,

$$I_s = \sum_{b=1}^{N_u} w_b \int_{-\infty}^{t} d\tau \, K_s(t - \tau) \rho_b(\tau) \, . \tag{7.2}$$

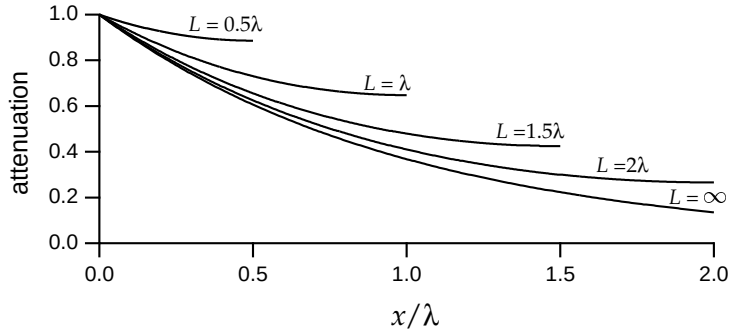

Figure 6.11 Voltage and current attenuation for the Rall model. The attenuation plotted is the ratio of the dendritic voltage to the somatic voltage for the recording setup of figure 6.10, or the ratio of the somatic current to the electrode current for the arrangement in figure 6.12. Attenuation is plotted as a function of x/λ for different equivalent cable lengths.

Figures 6.10 and 6.12 depict static solutions of the Rall model for two different recording configurations, expressed in the form of equivalent circuits. The equivalent circuits are an intuitive way of describing the solution of the cable equation. In figure 6.10, constant current is injected into the soma. The circuit diagram shows an arrangement of resistors that replicates the results of solving the time-independent cable equation (equation 6.17) for the purposes of voltage measurements at the soma, v_{soma}, and at a distance x along the equivalent cable, $v(x)$. The values for these resistances (and similarly the values of R_3 and R_4 given below) are set so that the equivalent circuit reconstructs the solution of the cable equation obtained using standard methods (see, for example, Tuckwell, 1988). R_{soma} is the membrane resistance of the soma, and

$$R_1 = \frac{R_\lambda \left(\cosh\left(L/\lambda\right) - \cosh\left((L-x)/\lambda\right)\right)}{\sinh\left(L/\lambda\right)} \tag{6.23}$$

$$R_2 = \frac{R_\lambda \cosh\left((L-x)/\lambda\right)}{\sinh\left(L/\lambda\right)}. \tag{6.24}$$

Expressions for v_{soma} and $v(x)$, arising directly from the equivalent circuit using standard rules of circuit analysis (see the Mathematical Appendix), are given at the right side of figure 6.10.

The input resistance of the Rall model neuron, as measured from the soma, is determined by the somatic resistance R_{soma} acting in parallel with the effective resistance of the cable, and is $(R_1 + R_2)R_{\text{soma}}/(R_1 + R_2 + R_{\text{soma}})$. The effective resistance of the cable, $R_1 + R_2 = R_\lambda / \tanh(L/\lambda)$, approaches the value R_λ when $L \gg \lambda$. The effect of lengthening a cable saturates when it gets much longer than its electrotonic length. The voltage attenuation caused by the cable is defined as the ratio of the dendritic potential to the somatic potential, and in this case it is given by

$$\frac{v(x)}{v_{\text{soma}}} = \frac{R_2}{R_1 + R_2} = \frac{\cosh\left((L-x)/\lambda\right)}{\cosh\left(L/\lambda\right)}. \tag{6.25}$$

typically constructed, an action potential fired by the model unit duplicates the effect of all the neurons it represents firing synchronously. Not surprisingly, such models tend to exhibit large-scale synchronization unlike anything seen in a healthy brain.

Firing-rate models also have their limitations. They cannot account for aspects of spike timing and spike correlations that may be important for understanding nervous system function. Firing-rate models are restricted to cases where the firing of neurons in a network is uncorrelated, with little synchronous firing, and where precise patterns of spike timing are unimportant. In such cases, comparisons of spiking network models with models that use firing-rate descriptions have shown that they produce similar results. Nevertheless, the exploration of neural networks undoubtedly requires the use of both firing-rate and spiking models.

7.2 Firing-Rate Models

As discussed in chapter 1, the sequence of spikes generated by a neuron is completely characterized by the neural response function $\rho(t)$, which consists of δ function spikes located at times when the neuron fired action potentials. In firing-rate models, the exact description of a spike sequence provided by the neural response function $\rho(t)$ is replaced by the approximate description provided by the firing rate $\mathsf{r}(t)$. Recall from chapter 1 that $\mathsf{r}(t)$ is defined as the probability density of firing and is obtained from $\rho(t)$ by averaging over trials. The validity of a firing-rate model depends on how well the trial-averaged firing rate of network units approximates the effect of actual spike sequences on the network's dynamic behavior.

The replacement of the neural response function by the corresponding firing rate is typically justified by the fact that each network neuron has a large number of inputs. Replacing $\rho(t)$, which describes an actual spike train, with the trial-averaged firing rate $\mathsf{r}(t)$ is justified if the quantities of relevance for network dynamics are relatively insensitive to the trial-to-trial fluctuations in the spike sequences represented by $\rho(t)$. In a network model, the relevant quantities that must be modeled accurately are the total inputs for the neurons within the network. For any single synaptic input, the trial-to-trial variability is likely to be large. However, if we sum the input over many synapses activated by uncorrelated presynaptic spike trains, the mean of the total input typically grows linearly with the number of synapses, while its standard deviation grows only as the square root of the number of synapses. Thus, for uncorrelated presynaptic spike trains, using presynaptic firing rates in place of the actual presynaptic spike trains may not significantly modify the dynamics of the network. Conversely, a firing-rate model will fail to describe a network adequately if the presynaptic inputs to a substantial fraction of its neurons are correlated. This can occur, for example, if the presynaptic neurons fire synchronously.

The synaptic input arising from a presynaptic spike train is effectively fil-

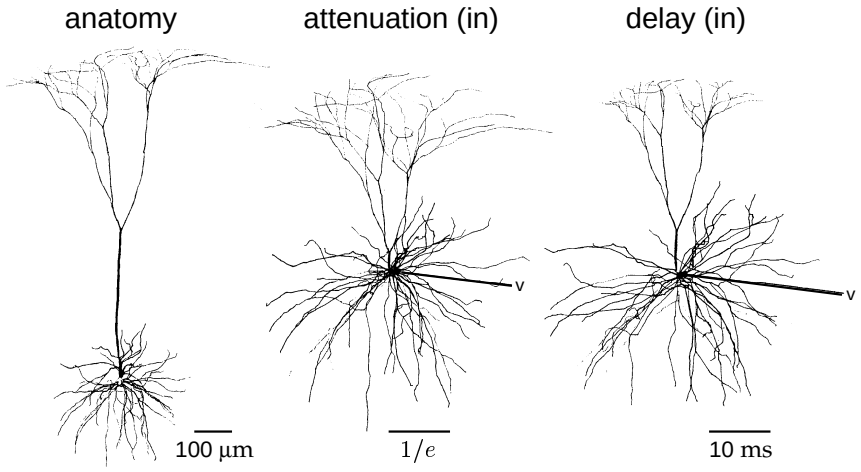

Figure 6.13 The morphoelectrotonic transform of a cortical neuron. The left panel is a normal drawing of the neuron. The central panel is a diagram in which the distance between any point and the soma is proportional to the logarithm of the steady-state attenuation between the soma and that point for static current injected at the terminals of the dendrites. The scale bar denotes the distance corresponding to an attenuation of $\exp(-1)$. In the right panel, the distance from the soma to a given point is proportional to the inward delay, which is the centroid of the soma potential minus the centroid at the periphery when a pulse of current is injected peripherally. The v labels in the diagrams indicate that the reference potential in these cases is the somatic potential. (Adapted from Zador et al, 1995.)

considered. Fortunately, efficient numerical schemes (discussed later in this chapter) exist for generating solutions for complex cable structures. However, even when the solution is known, it is still difficult to visualize the effects of a complex morphology on the potential. Zador et al. (1995; see also Tsai et al., 1994) devised a scheme for depicting the attenuation and delay of the membrane potential for complex morphologies. The voltage attenuation, as plotted in figure 6.11, is not an appropriate quantity to represent geometrically because it is not additive. Consider three points along a cable satisfying $x_1 > x_2 > x_3$. The attenuation between x_1 and x_3 is the product of the attenuation from x_1 to x_2 and from x_2 to x_3, $v(x_1)/v(x_3) = (v(x_1)/v(x_2))(v(x_2)/v(x_3))$. An additive quantity can be obtained by taking the logarithm of the attenuation, due to the identity $\ln(v(x_1)/v(x_3)) = \ln(v(x_1)/v(x_2)) + \ln(v(x_2)/v(x_3))$. The morphoelectrotonic transform is a diagram of a neuron in which the distance between any two points is determined by the logarithm of the ratio of the membrane potentials at these two locations, not by the actual size of the neuron.

morphoelectrotonic transform

Another morphoelectrotonic transform can be used to indicate the amount of delay in the voltage waveform produced by a transient input current. The morphoelectrotonic transform uses a definition of delay different from that used in Figure 6.8B. The delay between any two points is defined as the difference between the centroid, or center of "mass", of the voltage

7 Network Models

7.1 Introduction

Extensive synaptic connectivity is a hallmark of neural circuitry. For example, a typical neuron in the mammalian neocortex receives thousands of synaptic inputs. Network models allow us to explore the computational potential of such connectivity, using both analysis and simulations. As illustrations, we study in this chapter how networks can perform the following tasks: coordinate transformations needed in visually guided reaching, selective amplification leading to models of simple and complex cells in primary visual cortex, integration as a model of short-term memory, noise reduction, input selection, gain modulation, and associative memory. Networks that undergo oscillations are also analyzed, with application to the olfactory bulb. Finally, we discuss network models based on stochastic rather than deterministic dynamics, using the Boltzmann machine as an example.

cortical columns

Neocortical circuits are a major focus of our discussion. In the neocortex, which forms the convoluted outer surface of the (for example) human brain, neurons lie in six vertical layers highly coupled within cylindrical columns. Such columns have been suggested as basic functional units, and stereotypical patterns of connections both within a column and between columns are repeated across cortex. There are three main classes of interconnections within cortex, and in other areas of the brain as well.

feedforward, recurrent, and top-down connections

Feedforward connections bring input to a given region from another region located at an earlier stage along a particular processing pathway. Recurrent synapses interconnect neurons within a particular region that are considered to be at the same stage along the processing pathway. These may include connections within a cortical column as well as connections between both nearby and distant cortical columns within a region. Top-down connections carry signals back from areas located at later stages. These definitions depend on how the region being studied is specified and on the hierarchical assignment of regions along a pathway. In general, neurons within a given region send top-down projections back to the areas from which they receive feedforward input, and receive top-down input from the areas to which they project feedforward output. The numbers, though not necessarily the strengths, of feedforward and top-down

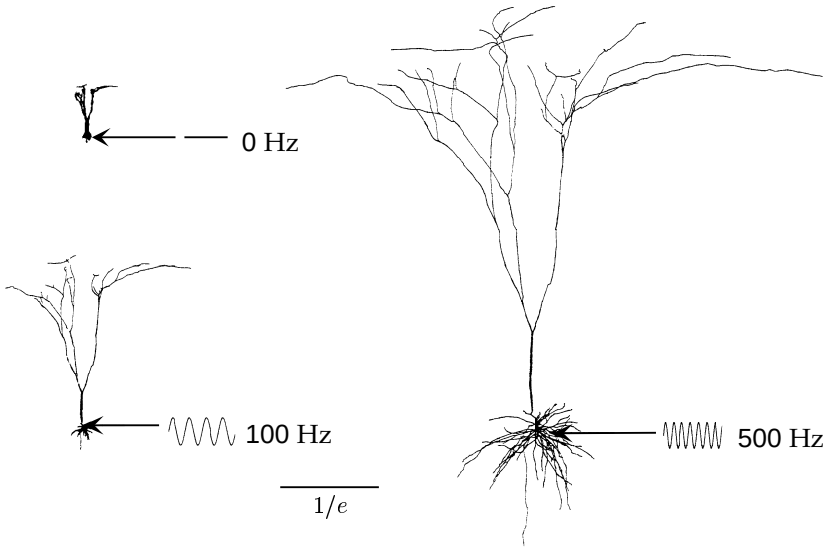

Figure 6.14 Morphoelectrotonic transforms of the same neuron as in figure 6.13 but showing the outward log-attenuation for constant and oscillating input currents. Distances in these diagrams are proportional to the logarithm of the amplitude of the voltage oscillations at a given point divided by the amplitude of the oscillations at the soma when a sinusoidal current is injected into the soma. The upper left panel corresponds to constant current injection, the lower left panel to sinusoidal current injection at a frequency of 100 Hz, and the right panel to an injection frequency of 500 Hz. The scale bar denotes the distance corresponding to an attenuation of $\exp(-1)$. (Adapted from Zador et al., 1995.)

number of compartments used and on their size relative to the length constants that characterize their electrotonic compactness. Figure 6.15 shows a schematic diagram of a cortical pyramidal neuron, along with a series of compartmental approximations of its structure. The number of compartments used can range from thousands, in some models, to one, for the description at the extreme right of figure 6.15.

In a multi-compartment model, each compartment has its own membrane potential V_μ (where μ labels compartments), and its own gating variables that determine the membrane current for compartment μ, $i_{\rm m}^\mu$. Each membrane potential V_μ satisfies an equation similar to 6.1 except that the compartments couple to their neighbors in the multi-compartment structure (figure 6.16). For a nonbranching cable, each compartment is coupled to two neighbors and the equations for the membrane potentials of the compartments are

$$c_{\rm m}\frac{dV_\mu}{dt} = -i_{\rm m}^\mu + \frac{I_{\rm e}^\mu}{A_\mu} + g_{\mu,\mu+1}(V_{\mu+1}-V_\mu) + g_{\mu,\mu-1}(V_{\mu-1}-V_\mu)\,. \quad (6.29)$$

Here $I_{\rm e}^\mu$ is the total electrode current flowing into compartment μ, and A_μ is its surface area. Compartments at the ends of a cable have only one neighbor, and thus only a single term replacing the last two terms in equation 6.29. For a compartment where a cable branches in two, there are

to determine ΔV_μ. To do this, we write $V_\mu(t + z\Delta t) \approx V_\mu(t) + z\Delta V_\mu$ and likewise for $V_{\mu\pm1}$. Substituting this into equation 6.48 gives

$$\Delta V_\mu = b_\mu \Delta V_{\mu-1} + c_\mu \Delta V_\mu + d_\mu \Delta V_{\mu+1} + f_\mu \,, \tag{6.49}$$

where

$$\begin{aligned} b_\mu &= B_\mu z\Delta t \,, \quad c_\mu = C_\mu z\Delta t \,, \quad d_\mu = D_\mu z\Delta t \,, \\ f_\mu &= (F_\mu + B_\mu V_{\mu-1}(t) + C_\mu V_\mu(t) + D_\mu V_{\mu+1}(t))\Delta t \,. \end{aligned} \tag{6.50}$$

Equation 6.49 for all μ values provides a set of coupled linear equations for the quantities ΔV_μ. An efficient method exists for solving these equations (Hines, 1984; Tuckwell, 1988). We illustrate the method for a single, nonbranching cable that begins at compartment $\mu = 1$, so that $b_1 = 0$, and ends at compartment $\mu = N$, so $d_N = 0$. The method consists of solving equation 6.49 for ΔV_μ in terms of $\Delta V_{\mu+1}$ sequentially, starting at one end of the cable and proceeding to the other end. For example, if we start the procedure at compartment 1, ΔV_1 can be expressed as

$$\Delta V_1 = \frac{d_1 \Delta V_2 + f_1}{1 - c_1}. \tag{6.51}$$

Substituting this into the equation 6.49 for $\mu = 2$ gives

$$\Delta V_2 = c_2' \Delta V_2 + d_2 \Delta V_3 + f_2' \,, \tag{6.52}$$

where $c_2' = c_2 + b_2 d_1/(1 - c_1)$ and $f_2' = f_2 + b_2 f_1/(1 - c_1)$. We now repeat the procedure going down the cable. At each stage, we solve for $\Delta V_{\mu-1}$ in terms of ΔV_μ, finding

$$\Delta V_{\mu-1} = \frac{d_{\mu-1} \Delta V_\mu + f_{\mu-1}'}{1 - c_{\mu-1}'} \,, \tag{6.53}$$

where

$$c_{\mu+1}' = c_{\mu+1} + \frac{b_{\mu+1} d_\mu}{1 - c_\mu'} \tag{6.54}$$

and

$$f_{\mu+1}' = f_{\mu+1} + \frac{b_{\mu+1} f_\mu'}{1 - c_\mu'} \,. \tag{6.55}$$

Finally, when we get to the end of the cable, we can solve for

$$\Delta V_N = \frac{f_N'}{1 - c_N'} \tag{6.56}$$

because $d_N = 0$.

The procedure for computing all the ΔV_μ is the following. Define $c_1' = c_1$ and $f_1' = f_1$ and iterate equations 6.54 and 6.55 down the length of the

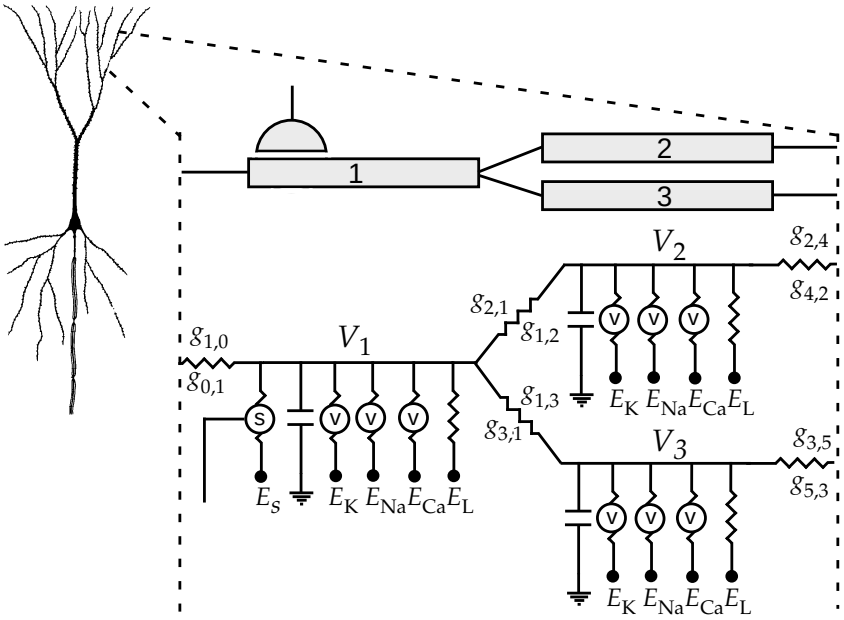

Figure 6.16 A multi-compartment model of a neuron. The expanded region shows three compartments at a branch point where a single cable splits into two. Each compartment has membrane and synaptic conductances, as indicated by the equivalent electrical circuit, and the compartments are coupled together by resistors. Although a single resistor symbol is drawn, note that $g_{\mu,\mu'}$ is not necessarily equal to $g_{\mu',\mu}$.

compartment μ, $2\pi a_\mu L_\mu$, which gives

$$g_{\mu,\mu'} = \frac{a_\mu a_{\mu'}^2}{r_L L_\mu (L_\mu a_{\mu'}^2 + L_{\mu'} a_\mu^2)}. \tag{6.30}$$

Equations 6.29 for all of the compartments of a model determine the membrane potential throughout the neuron with a spatial resolution given by the compartment size. An efficient method for integrating the coupled multi-compartment equations is discussed in appendix B. Using this scheme, models can be integrated numerically with excellent efficiency, even those involving large numbers of compartments. Such integration schemes are built into neuron simulation software packages such as Neuron and Genesis.

Action-Potential Propagation Along an Unmyelinated Axon

As an example of multi-compartment modeling, we simulate the propagation of an action potential along an unmyelinated axon. In this model, each compartment has the same membrane conductances as the single-compartment Hodgkin-Huxley model discussed in chapter 5. The different compartments are joined together in a single nonbranching cable

Transient Ca^{2+} Conductance

The gating functions used for the variables M and H in the transient Ca^{2+} conductance model we discussed, with V in units of mV and τ_M and τ_H in ms, are

$$M_\infty = \frac{1}{1 + \exp\left(-(V + 57)/6.2\right)} \tag{6.38}$$

$$H_\infty = \frac{1}{1 + \exp\left((V + 81)/4\right)} \tag{6.39}$$

$$\tau_M = 0.612 + \left(\exp\left(-(V + 132)/16.7\right) + \exp\left((V + 16.8)/18.2\right)\right)^{-1} \tag{6.40}$$

and

$$\tau_H = \begin{cases} \exp\left((V + 467)/66.6\right) & \text{if } V < -80 \text{ mV} \\ 28 + \exp\left(-(V + 22)/10.5\right) & \text{if } V \geq -80 \text{ mV}\,. \end{cases} \tag{6.41}$$

Ca^{2+}-dependent K^+ Conductance

The gating functions used for the Ca^{2+}-dependent K^+ conductance we discussed, with V in units of mV and τ_c in ms, are

$$c_\infty = \left(\frac{[Ca^{2+}]}{[Ca^{2+}] + 3\mu\text{M}}\right) \frac{1}{1 + \exp(-(V + 28.3)/12.6)} \tag{6.42}$$

and

$$\tau_c = 90.3 - \frac{75.1}{1 + \exp(-(V + 46)/22.7)}\,. \tag{6.43}$$

B: Integrating Multi-compartment Models

Multi-compartment models are defined by a coupled set of differential equations (equation 6.29), one for each compartment. There are also gating variables for each compartment, but these involve only the membrane potential (and possibly Ca^{2+} concentration) within that compartment, and integrating their equations can be handled as in the single-compartment case using the approach discussed in appendix B of chapter 5. Integrating the membrane potentials for the different compartments is more complex because they are coupled to each other.

Equation 6.29, for the membrane potential within compartment μ, can be written in the form

$$\frac{dV_\mu}{dt} = B_\mu V_{\mu-1} + C_\mu V_\mu + D_\mu V_{\mu+1} + F_\mu\,, \tag{6.44}$$

Refractoriness following spiking has a number of other consequences for action-potential propagation. Two action potentials moving in opposite directions that collide annihilate one another because they cannot pass through each other's trailing refractory regions. Refractoriness also keeps action potentials from reflecting off the ends of axon cables, which avoids the impedance matching needed to prevent reflection from the ends of ordinary electrical cables.

The propagation velocity for an action potential along an unmyelinated axon is proportional to the ratio of the electrotonic length constant to the membrane time constant, $\lambda/\tau_m = (a/(2c_m^2 r_L r_m))^{1/2}$. This is proportional to the square root of the axon radius. The square-root dependence of the propagation speed on the axon radius means that thick axons are required to achieve high action-potential propagation speeds, and the squid giant axon is an extreme example. Action-potential propagation can also be sped up by covering the axon with an insulating myelin wrapping, as we discuss next.

Action-Potential Propagation Along a Myelinated Axon

saltatory propagation

Many axons in vertebrates are covered with an insulating sheath of myelin except at gaps, called the nodes of Ranvier, where there is a high density of fast voltage-dependent Na^+ channels (see figure 6.18A). The myelin sheath consists of many layers of glial cell membrane wrapped around the axon. This gives the myelinated region of the axon a very high membrane resistance and a small membrane capacitance. This results in what is called saltatory propagation, in which membrane potential depolarization is transferred passively down the myelin-covered sections of the axon, and action potentials are actively regenerated at the nodes of Ranvier. Figure 6.18A shows an equivalent circuit for a multi-compartment model of a myelinated axon.

We can compute the capacitance of a myelin-covered axon by treating the myelin sheath as an extremely thick cell membrane. Consider the geometry shown in the cross-sectional diagram of figure 6.18B. The myelin sheath extends from the radius a_1 of the axon core to the outer radius a_2. For calculational purposes, we can think of the myelin sheath as being made of a series of thin, concentric cylindrical shells. The capacitances of these shells combine in series to make up the full capacitance of the myelinated axon. If a single layer of cell membrane has thickness d_m and capacitance per unit area c_m, the capacitance of a cylinder of membrane of radius a, thickness Δa, and length L is $c_m 2\pi d_m L a/\Delta a$. According to the rule for capacitors in series, the inverse of the total capacitance is obtained by adding the inverses of the individual capacitances. The capacitance of a myelinated cylinder of length L and the dimensions in figure 6.18B is then obtained by taking the limit $\Delta a \to 0$ and integrating,

$$\frac{1}{C_m} = \frac{1}{c_m 2\pi d_m L}\int_{a_1}^{a_2}\frac{da}{a} = \frac{\ln(a_2/a_1)}{c_m 2\pi d_m L}. \qquad (6.31)$$

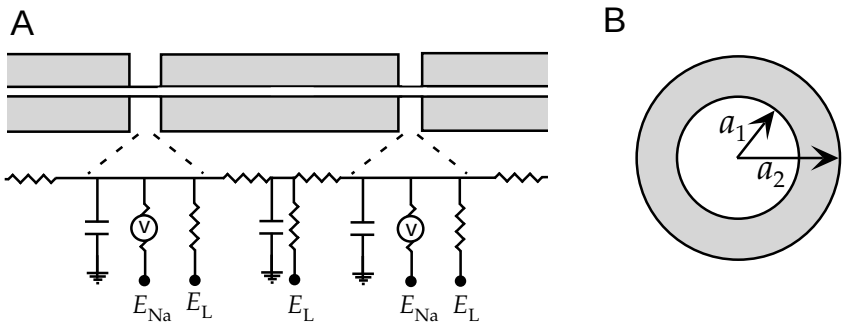

Figure 6.18 A myelinated axon. (A) The equivalent circuit for a multi-compartment representation of a myelinated axon. The myelinated segments are represented by a membrane capacitance, a longitudinal resistance, and a leakage conductance. The nodes of Ranvier also contain a voltage-dependent Na^+ conductance. (B) A cross section of a myelinated axon consisting of a central axon core of radius a_1 and a myelin sheath making the outside radius a_2.

A re-evaluation of the derivation of the linear cable equation earlier in this chapter indicates that the equation describing the membrane potential along the myelinated sections of an axon, in the limit of infinite resistance for the myelinated membrane and with $i_e = 0$, is

$$\frac{C_m}{L}\frac{\partial v}{\partial t} = \frac{\pi a_1^2}{r_L}\frac{\partial^2 v}{\partial x^2}. \tag{6.32}$$

This is equivalent to the diffusion equation, $\partial v/\partial t = D\partial^2 v/\partial x^2$, with diffusion constant $D = \pi a_1^2 L/(C_m r_L) = a_1^2 \ln(a_2/a_1)/(2c_m r_L d_m)$. It is interesting to compute the inner core radius, a_1, that maximizes this diffusion constant for a fixed outer radius a_2. Setting the derivative of D with respect to a_1 to 0 gives the optimal inner radius $a_1 = a_2 \exp(-1/2)$ or $a_1 \approx 0.6a_2$. An inner core fraction of 0.6 is typical for myelinated axons. This indicates that for a given outer radius, the thickness of myelin maximizes the diffusion constant along the myelinated axon segment.

At the optimal ratio of radii, $D = a_2^2/(4ec_m r_L d_m)$, which is proportional to the square of the axon radius. Because of the form of the diffusion equation it obeys with this value of D, v can be written as a function of x/a_2 and t. This scaling implies that the propagation velocity for a myelinated cable is proportional to a_2, that is, to the axon radius, not its square root (as in the case of an unmyelinated axon). Increasing the axon radius by a factor of 4, for example, increases the propagation speed of an unmyelinated cable only by a factor of 2, while it increases the speed for a myelinated cable fourfold.